

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF RADIO ELECTRONICS

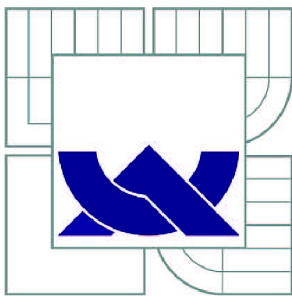
MĚŘENÍ KVALITY A DOSTUPNOSTI GSM SÍŤE

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

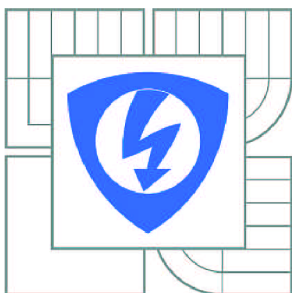
Bc. MAREK VONDRA

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ

ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF RADIO ELECTRONICS

## MĚŘENÍ KVALITY A DOSTUPNOSTI GSM SÍTĚ

QUALITY AND ACCESSABILITY MEASUREMENT OF GSM NETWORK

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

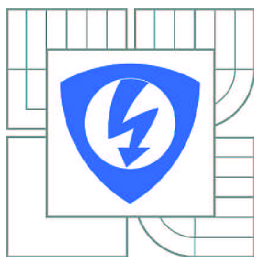
Bc. MAREK VONDRA

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. STANISLAV HANUS, CSc.

BRNO 2011



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav radioelektroniky

# Diplomová práce

magisterský navazující studijní obor  
**Elektronika a sdělovací technika**

**Student:** Bc. Marek Vondra

**ID:** 101884

**Ročník:** 2

**Akademický rok:** 2010/2011

## NÁZEV TÉMATU:

### Měření kvality a dostupnosti GSM sítě

#### POKYNY PRO VYPRACOVÁNÍ:

Prostudujte možnosti měření kvality a dostupnosti mobilní sítě. Dále prostudujte různé definice kritérií kvality a jiných metrik (dostupnost, úspěšnost připojení, propustnost apod.). Navrhněte systém, který umožní pravidelné měření kvality a dostupnosti mobilní sítě GSM. Navržený systém ověřte v laboratorních, případně v reálných podmínkách mobilní sítě GSM.

#### DOPORUČENÁ LITERATURA:

[1] MORROW, R. Wireless Network Coexistence. The McGraw-Hill Companies, 2004.

[2] STAVROULAKIS, P. Interference Analysis and Reduction for Wireless Systems. Artech House Inc., 2004.

**Termín zadání:** 7.2.2011

**Termín odevzdání:** 20.5.2011

**Vedoucí práce:** prof. Ing. Stanislav Hanus, CSc.

**prof. Dr. Ing. Zbyněk Raida**

*Předseda oborové rady*

#### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Diplomová práce je zaměřena na možnosti měření kvality a dostupnosti mobilní sítě a také na hodnocení kvality, úspěšnosti připojení, propustnosti apod. Obsahuje stručný popis základních mobilních a komunikačních technologií. Dále pak rozbor možností měření mobilní sítě a možnosti, jak měření realizovat. A také obsahuje návrh programu pro měření a vyhodnocování sítě GSM, pomocí několika testů, připojeným mobilním telefonem přes Bluetooth.

## **KLÍČOVÁ SLOVA**

Release, GSM, Mobilní stanice, GPRS, EDGE, EGPRS, CDMA, UMTS, WiMAX, LTE, QVoice, TEMS, AT příkazy, Borland C++ Builder, ComPort Library, RAS Manager, ICS - Internet Component Suite, HTTPGet

## **ABSTRACT**

The thesis is specialized on a possibility of measuring qualities and availability of a cellular radio system and also on evaluation qualities, fruitfulness interface, permeability etc. It includes short description of a basic mobile technology. Further analysis of the possibility of measuring the mobile network and how implement the measure. It also includes a draft program for measuring and evaluation of the GSM network with help several tests by a mobile phone connected through the Bluetooth.

## **KEYWORDS**

Release, GSM, Mobile Station, GPRS, EDGE, EGPRS, CDMA, UMTS, WiMAX, LTE, QVoice, TEMS, AT command, Borland C++ Bilder, ComPort Library, RAS Manager, ICS - Internet Component Suite, HTTPGet

VONDRA, M. *Měření kvality a dostupnosti GSM sítě*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 59 s., 2 příl. Vedoucí diplomové práce prof. Ing. Stanislav Hanus, CSc..

## Prohlášení

Prohlašuji, že svou diplomovou práci na téma „Měření kvality a dostupnosti GSM sítě“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 20. května 2011

.....  
podpis autora

## Poděkování

Děkuji vedoucímu diplomové práce Prof. Ing. Stanislavu Hanusovi, CSc. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mého semestrálního projektu.

V Brně dne 20. května 2011

.....  
podpis autora

# OBSAH

Seznam obrázků.....	vii
Seznam tabulek.....	ix
<b>1 Úvod.....</b>	<b>1</b>
<b>2 Technologie v mobilních sítích.....</b>	<b>2</b>
2.1 GSM.....	2
2.1.1 Struktura sítě GSM .....	3
2.1.2 Mobilní stanice .....	5
2.2 GPRS .....	6
2.2.1 Třídy zařízení GPRS .....	6
2.2.2 Kódová schémata .....	6
2.2.3 Možnosti mobilních zařízení .....	7
2.2.4 Struktura GPRS.....	7
2.3 EDGE.....	8
2.4 EGPRS .....	9
2.5 CDMA .....	10
2.6 UMTS .....	11
2.6.1 Struktura sítě.....	11
2.6.2 Režimy oddělení uplinku a downlinku TDD a FDD .....	12
2.7 WiMAX .....	13
2.8 Bluetooth.....	13
2.8.1 Počátky a verze technologie Bluetooth.....	13
2.8.2 Struktura a vlastnosti technologie a komunikace .....	14
2.8.3 Bezpečnost .....	15
<b>3 Měření parametrů sítě GSM .....</b>	<b>16</b>
3.1 Parametry sítě GSM a měřicí metody.....	16
3.1.1 OMC .....	16
3.1.2 Call trace .....	17
3.1.3 Drive test.....	17
3.1.4 Měření kvality řeči .....	17
3.2 Měřicí systém QVoice .....	18

3.2.1	QVoice Smart .....	18
3.2.2	QVoice Companion .....	19
3.2.3	QVoice Compact.....	19
3.2.4	QVoice Symphony.....	19
3.2.5	QVoice Symphony Multi.....	20
3.2.6	QVoice Presentation .....	20
3.3	Měřicí systém Tems.....	20
3.3.1	TEMS Pocket.....	21
3.3.2	TEMS Investigation.....	21
3.3.3	TEMS Automatic.....	21
3.3.4	TEMS Software .....	21
<b>4</b>	<b>Měření parametrů sítě GSM dle požadavků uživatele.....</b>	<b>22</b>
4.1	Požadavky na řešení.....	22
4.2	Postup v řešení .....	23
4.2.1	Prostředí Matlab.....	23
4.2.2	Výběr mobilního telefonu.....	24
4.2.3	Java aplikace na MS .....	25
4.2.4	GSM moduly.....	26
4.3	Zvolené řešení .....	27
4.3.1	Postup a předpoklady.....	27
4.3.2	Ovládací software .....	28
<b>5</b>	<b>Program pro měření kvality a dostupnosti GSM sítě.....</b>	<b>30</b>
5.1	Verze programu .....	30
5.2	Jednotlivé části programu.....	31
5.2.1	Záložka Připojení zařízení .....	31
5.2.2	Záložka Testování.....	33
5.2.3	Záložka Plánování testů.....	34
5.2.4	Záložka Databáze.....	36
5.2.5	Záložka Grafy .....	37
5.2.6	Záložka O programu .....	38
5.2.7	Nápověda .....	39
5.3	Testovací prostředky.....	40
5.3.1	Test DSS – Doba sestavení spojení .....	40
5.3.2	Test DOS – Doba odeslání SMS.....	42
5.3.3	Test PIN – PING.....	43



5.3.4	Test DPW – Doba přístupu WWW.....	45
5.3.5	Test PRD – Přenosová rychlost (Dial-up) .....	46
5.3.6	Vlastní test .....	47
5.4	Chyby programu .....	49
<b>6</b>	<b>Naměřené a získané hodnoty.....</b>	<b>50</b>
6.1	Test DSS .....	50
6.2	Test DOS.....	50
6.3	Test PIN .....	51
6.4	Test DPW.....	52
6.5	Test PRD.....	52
<b>7</b>	<b>Závěr.....</b>	<b>54</b>
	<b>Literatura.....</b>	<b>56</b>
	<b>Seznam symbolů, veličin a zkratk.....</b>	<b>57</b>
	<b>Seznam příloh.....</b>	<b>60</b>

# SEZNAM OBRÁZKŮ

Obr. 2.1: Logo GSM [2] .....	3
Obr. 2.2: Struktura sítě GSM [3] .....	3
Obr. 2.3: Rozložení buněk ve svazku [3].....	4
Obr. 2.4: Blokové schéma mobilní stanice [4] .....	5
Obr. 2.5: Schéma sítě GSM s GPRS s rozepsanými anglickými zkratkami [6] .....	8
Obr. 2.6: Konstelační diagram 8-PSK .....	9
Obr. 2.7: Znázornění přístupu TDMA a CDMA [8].....	10
Obr. 2.8: Architektura UMTS s vyznačením pomocí zkratek (Release 99) [9] .....	11
Obr. 2.9: Rozložení kmitočtů UMTS [10].....	12
Obr. 2.10: Logo Bluetooth [13] .....	13
Obr. 3.1: QVoice Smart .....	18
Obr. 3.2: QVoice Companion .....	19
Obr. 3.3: QVoice Compact .....	19
Obr. 3.4: QVoice Symphony .....	20
Obr. 3.5: QVoice Symphony Multi .....	20
Obr. 4.1: Ukázka grafického prostředí GUI .....	24
Obr. 4.2: GSM modul TM2 [16].....	26
Obr. 4.3: GPRS/GSM modem Wismo Quick [16] .....	27
Obr. 4.4: Náhled vizuálního programu posílajícího AT příkazy .....	29
Obr. 5.1: Náhled na záložku Připojení zařízení .....	33
Obr. 5.2: Náhled na záložku Testování.....	34
Obr. 5.3: Náhled na záložku Plánování testů.....	35
Obr. 5.4: Náhled na záložku Databáze .....	37
Obr. 5.5: Náhled na záložku Grafy .....	38
Obr. 5.6: Náhled na záložku O programu.....	39
Obr. 5.7: Test - Doba sestavení spojení .....	42
Obr. 5.8: Test - Doba odeslání SMS .....	43
Obr. 5.9: Test - PING.....	45
Obr. 5.10: Test - Doba přístupu WWW .....	46
Obr. 5.11: Test - Přenosová rychlost (Dial-up) .....	47
Obr. 5.12: Test - Vlastní test.....	48

Obr. 6.1: Výstup testu DSS.....	50
Obr. 6.2: Výstup testu DOS.....	51
Obr. 6.3: Výstup testu PIN.....	51
Obr. 6.4: Výstup testu DPW.....	52
Obr. 6.5: Výstup testu PRD.....	53

## SEZNAM TABULEK

Tab. 2.1: Přehled vydání standardu 3GPP [1] .....	2
Tab. 2.2: Maximální výkonové úrovně pro dané výkonnostní třídy .....	6
Tab. 2.3: Kódová schémata GPRS [5].....	7
Tab. 2.4: Přehled využití timeslotů podle podporované třídy MS [5] .....	7
Tab. 2.5: Kódová schémata EGPRS [7] .....	9

# 1 ÚVOD

Postupem techniky a rozvojem bezdrátových sítí vzrůstá i počet uživatelů, kteří využívají mobilní komunikaci, byť jde o pouhé posílání SMS zpráv nebo surfování na internetu. Snahou operátorů je tedy zkvalitnit svoje služby natolik, aby byly pro daného zákazníka co nejkomfortnější. Tato diplomová práce se však nezabývá nabídkou služeb jednotlivých operátorů, ale bezdrátovou sítí GSM, která služby zprostředkovává, jak již název diplomové práce „Měření kvality a dostupnosti GSM sítě“ napovídá.

Diplomová práce je zaměřena na rozbor měřících technik a prostředků, kterými jej lze realizovat. Výsledkem rozboru je stanovení požadavků na měřící program, který bude samostatně měřit, pomocí různých testů, parametry sítě GSM. Nejedná se však o měření parametrů z pohledu technologie sítě, ale z pohledu využitelnosti pro koncového uživatele. V očích uživatele je prvotní, aby služby (zprostředkovávané sítí) byly co nejpřívětivější a neznepříjemňovali uživateli život. Jedná se tedy o zhodnocení mobilní sítě z pohledu běžného uživatele.

V nynější době existuje několik komerčních prostředků, jako je QVoice nebo TEMS, pro měření parametrů bezdrátové sítě, ale jejich pořízení je poměrně nákladné a jejich ovládání je určeno spíše do odborných rukou. Vhodným doplňkem by tedy měla být vytvářená aplikace v této práci, která by měla vynikat jednoduchostí a přitom efektivností. Její vývoj však není cílem, aby konkurovala komerčním systémům a ani jiným.

Zaměření aplikace je striktně určeno pro datovou komunikaci a nikoliv hlasovou. Zaměřit se tedy lze na odesílání textových zpráv SMS, doby načítání WWW stránek, stahování souborů nebo doby potřebné k sestavení hovoru, což by bylo spíše zařaditelné na okraj hlasové komunikace.

Pro ucelený náhled na mobilní síť je v druhé kapitole uveden stručný přehled a rozbor používaných mobilních technologií, jako jsou GSM, GPRS, EDGE, CDMA, UMTS nebo i WiMAX, ač poslední není pro všechny uživatele příliš aktuální. Pro doplnění je uvedena technologie Bluetooth, protože přes rozhraní Bluetooth komunikuje vytvářená aplikace a mobilní zařízení sloužící ke komunikaci se sítí.

Třetí kapitola se již zabývá možnostmi měření parametrů mobilních sítí, měřícím systémem QVoice a TEMS a definicemi parametrů sítí. K tomu lze v této kapitole nalézt část, která se věnuje vlastnostem sítě z pohledu uživatele a také možnost srovnání více použitých operátorů zároveň. Nabízí se tak možnost porovnávat použité technologie a jejich vliv na kvalitu služeb přičemž požadavky a okolní vlivy budou téměř totožné.

Následující kapitola obsahuje rozbor a stanovení požadavků, které by měl splňovat navrhovaný systém. Možností se zdá být hned několik, jak stanovených cílů dosáhnout a proto je ve čtvrté kapitole popsán postup práce a slepé uličky, kterými se hledání řešení někdy ubíralo.

Popis vytvářené aplikace je uveden v páté kapitole, kde je i vysvětlen vlastní zdrojový kód jednotlivých částí a práce s jednotlivými částmi programu.

Poslední šestá kapitola završuje přehled naměřených parametrů při testování a vytváření programu.

## 2 TECHNOLOGIE V MOBILNÍCH SÍTÍCH

V dnešní době je již několik generací mobilních technologií. Tato kapitola se nebude zabývat prvními, které ještě byly analogové, ale některými zbývajícími od druhé generace (2G), které již byly digitální a které bylo možné využít pro datové služby. Vývoj je definován následujícím způsobem, uvedeným v tab. 2.1, kde jsou na začátku počáteční fáze a dále pak další vydání.

Tab. 2.1: Přehled vydání standardu 3GPP [1]

Verze	Vydání	Popis
Phase 1	1990	vlastnosti GSM
Phase 2	1995	vlastnosti GSM, kodek EFR
Release 96	1997 1. čtvrtletí	vlastnosti GSM, uživatelská rychlost přenosu 14,4 kbit/s
Release 97	1998 1. čtvrtletí	vlastnosti GSM, GPRS
Release 98	1998	vlastnosti GSM, AMR, EDGE, GPRS pro PCS1900
Release 99 Release 3	2000 1. čtvrtletí	první specifikace UMTS sítě 3G zahrnující vzduchové rozhraní CDMA
Release 4	2001 2. čtvrtletí	původně Release 2000, přidáno all-IP
Release 5	2002 1. čtvrtletí	představeno IMS a HSDPA (downlink 14,4 Mbit/s)
Release 6	2004 4. čtvrtletí	HSUPA (uplink 5,76 Mbit/s), MBMS (Multimedia broadcast multicast), vylepšení IMS – PoC, GAN, provoz Wireless LAN
Release 7	2007 4. čtvrtletí	zaměřeno na snížení latence, zlepšení QoS a real-time aplikací, HSPA +, možnost plateb přes MS, EDGE Evolution
Release 8	prosinec 2008	LTE (Long Term Evolution), HSPA Evolution, předeclarování UMTS z Release 4
Release 9	prosinec 2009	součinnost WiMAX a LTE/UMTS
Release 10	2010	LTE Advanced
Release 11	2010	Advanced IP – propojení servisních služeb, propojení mezi operátory jako poskytovatele aplikací třetích stran

Datum uvedení jednotlivých vydání (Release) jsou určeny datem představení, protože dokumentace obsahuje několik verzí různých vydání.

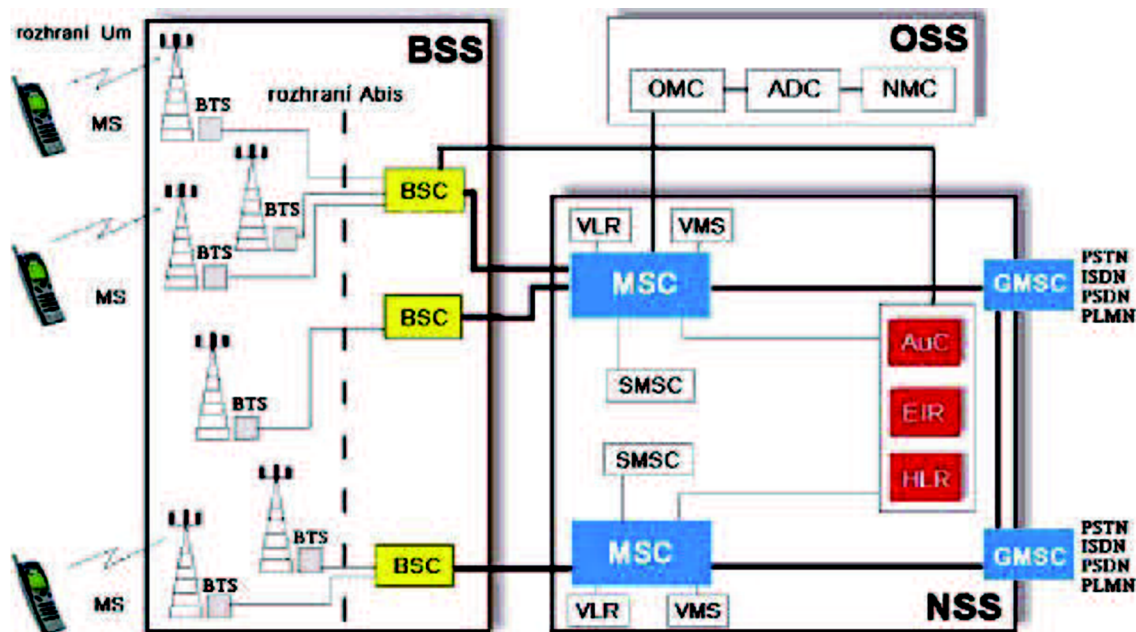
### 2.1 GSM

Základní technologií nynějších mobilních služeb je otevřený standard (vyvíjen 3GPP) GSM (Global System for Mobile Communication nebo také původně z francouzského Groupe Spécial Mobile), jde o systém druhé generace (2G). Na rozdíl od svých předchůdců se jedná o digitální systém a v průběhu dalšího vývoje do něj bylo možno implementovat datové přenosy (od roku 1997), tj. GPRS a EDGE. První specifikace GSM je z roku 1990. Systém je celosvětově rozšířen, v USA pod PCS 1900 pracujíc na jiných frekvencích. Logo GSM je na obr. 2.1.

Obr. 2.1: Logo GSM [2]

### 2.1.1 Struktura sítě GSM

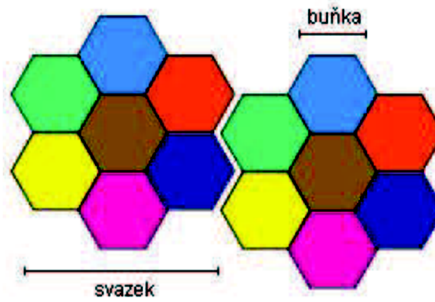
Struktura sítě GSM je na obr. 2.2, kde jsou zobrazeny jednotlivé součásti sítě.



Obr. 2.2: Struktura sítě GSM [3]

Struktura je složena ze tří subsystémů. Jsou to BSS, NSS a OSS.

První je subsystém základových stanic **BSS** (Base Station Subsystem), který je tvořen ze základnové stanice BTS (Base Transceiver Station) a základnové řídicí jednotky BSC (Base Station Controller). Se subsystémem NSS je BSS spojeno pomocí rozhraní Abis s přenosovou rychlostí 16 kbit/s. **BSC** může obsahovat jeden svazek 9 buněk (viz obr. 2.3) nebo svazek s menším počtem buněk. V každé buňce je umístěna buď jedna stanice BTS nebo stanic BTS může být více (pokud je použit systém sektorizace). BTS společně s BSC vytváří topologii sítě (hvězdicová, kaskádní, stromová) a jsou povětšinou propojeny radioreléovými spoji, ale může být spojení realizováno pomocí metalických nebo optických tras jako při propojení BSC a MSC (rozhraní A). Jednotky BSC také řídí dynamické přidělování kanálů a částečně plní přepojovací funkci. **BTS** umožňuje připojení MS ve formě rádiového spoje Um nebo I/F, udržuje a monitoruje spojení s MS. Základnové stanice se rozdělují do osmi výkonnostních tříd: 2,5; 5; 10; 20; 40; 80; 160; 320 W.



Obr. 2.3: Rozložení buněk ve svazku [3]

Druhý je síťový spojovací subsystém NSS (Network Switching Subsystem) tvořený částmi:

- mobilní radiotelefonní ústředna MSC (Mobile Switching Centre)
- domovský lokační registr HLR (Home Location Register)
- návštěvnický lokační registr VLR (Visitor Location Register)
- centrum autentičnosti VaC (Authentication Centre)
- identifikační registr mobilních stanic EIR (Equipment Identity Register)

K přenosu signalizace mezi jednotlivými prvky je použita signalizační síť SS7. Mezi jednotlivými prvky jsou různá rozhraní:

- rozhraní B mezi MSC a VLR
- rozhraní C mezi MSC a HLR
- rozhraní D mezi HLR a VLR
- rozhraní E mezi dvěma MSC
- rozhraní F mezi MSC a EIR
- rozhraní G mezi HLR a AuC

NSS realizuje mnoho specifických funkcí, které obstarávají jeho součásti. Funkce vycházejí z faktu, že se jedná o spojování mobilních účastníků. Hlavní funkce je však spojovací.

**MSC** má za úkol propojovat účastníky mezi sebou a případně je propojovat k externím sítím, jako jsou jiní operátoři nebo pevná linka. Aby toto bylo umožněno, musí být MSC typu Gateway obsahující jednotku IWF (Inter-Working Functionality). **HLR** plní funkci hlavní databáze, ve které jsou uloženy všechny důležité údaje o účastnících v síti. Obsahem registru jsou důležitá čísla IMSI (International Mobile Subscriber Identity), MSISDN (Mobile Subscriber ISDN Number), zpřístupněné služby a dále i údaje týkající se polohy uživatele. Každý operátor má minimálně jeden domovský lokační registr HLR. Součástí HLR je i centrum autentičnosti **AuC**, což je chráněná databáze, která obsahuje klíče pro ověření totožnosti účastníků. Dále se AuC stará o šifrovací klíč, kterým se šifruje každý účastnický signál, který je přenášený. Šifrovací klíč je unikátní pro každého účastníka a mění se v čase. Na registr HLR je napojen registr **EIR**, který obsahuje čísla mobilních stanic IMEI (International Mobile Equipment Identity), které jsou autorizovány v dané síti, také čísla kradených mobilních MS a stanic, které jsou porouchané nebo nesplňují jiné



parametry. Poslední částí je registr **VLR**, který přechodně uchovává a obnovuje data o uživateli, jež jsou v dosahu dané MSC. Obsah registru je podobný HLR, data jsou však vymazána, jakmile účastník opustí danou oblast.

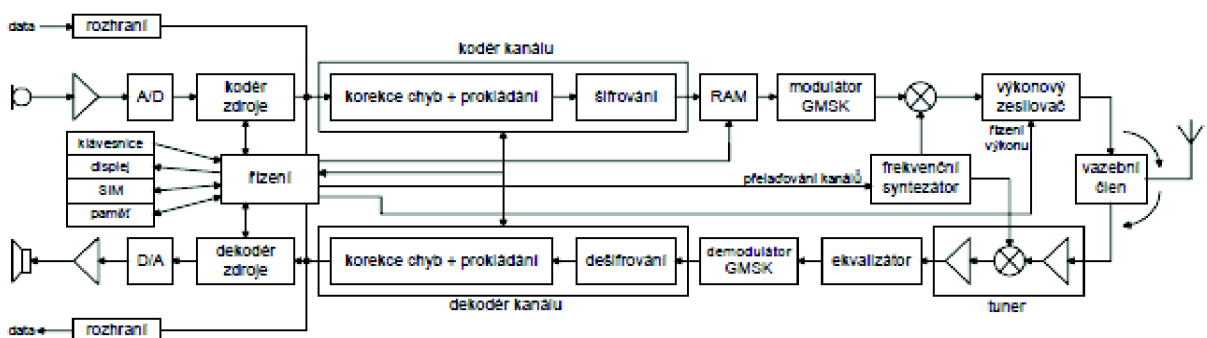
Třetím je operační a podpůrný subsystém **OSS** (Operational and Support Subsystem). OSS se skládá ze tří částí, kterými jsou: provozní a servisní centrum OMC (Operational and Maintenance Centre), centrum pro řízení sítě NMC (Network Management Centre) a administrativní centrum ADC (Administrative Centre). Celý subsystém má za úkol provádět management mobilních stanic (hledá porouchané MS), dále pak provádí dohled nad konfigurací sítě a údržbu sítě a také administraci účastníků (registrace, tarifkace apod.).

## 2.1.2 Mobilní stanice

Ačkoliv by tato kapitola mohla být samostatná, protože mobilní telefon je úzce spjat se všemi popisovanými technologiemi a využívá je, je zde zařazena, jelikož některé základní parametry mobilního telefonu byly definovány s počátkem GSM a postupně se s ním i vyvíjely.

Podle specifikace GSM je mobilní stanice celek dvou částí a to mobilního zařízení (přijímače/vysílače) a SIM (Subscriber Identification Module).

Mobilní zařízení je jednoznačně identifikováno podle vlastního čísla IMEI.



Obr. 2.4: Blokové schéma mobilní stanice [4]

Z blokového schéma mobilního telefonu, obr. 2.4, je patrné, že krom uživatelského rozhraní, jako je klávesnice, display, paměť a datové rozhraní, se mobilní stanice skládá i z převodníků pro převod analogového signálu (lidský hlas) na digitální a zpět, kodérů zdroje, korekce chyb, šifrování, modulátoru a demodulátoru, výstupního zesilovače a antény, přičemž většina prvků je řízena mikroprocesorem.

Druhou částí mobilní stanice je SIM, bez které přístroj nelze použít, pokud není zadána volba 112. SIM je přenosná a uchovává specifické informace o majiteli, jako je čtyřmístné číslo PIN (Personal Identification Number), neměnitelné číslo PUK (Personal Unblocking Key), IMSI, Autentifikační klíč (Ki), Šifrovací klíč (Kc), TMSI (Temporary Mobile Subscriber Identity), LAI (Location Area Identity). SIM slouží k ověření uživatele a ke službám, které může využívat.

Aby mobilní stanice nevysílala maximálním výkonem a nerušila tak okolí nebo nevysílala nízkým výkonem a neměla tak malý dosah, je stanoveno několik vysílacích výkonových úrovní, které by stanice měly splňovat, viz tab. 2.2. Vysílací výkon je za běžného provozu nastavován adaptivně.

Tab. 2.2: Maximální výkonové úrovně pro dané výkonnostní třídy

GSM Class	1	2	3	4	5
max. výkon [dBm]	43	39	37	33	29

## 2.2 GPRS

Datová služba GPRS (General Packet Radio Service) byla poprvé zahrnuta do standardu GSM v Release 97 a je označována jako 2,5G. Zavádí do GSM nový způsob přenosu dat, protože na rozdíl od staršího předchůdce CSD (Circuit Switched Data) nevyužívá přidělování maximálních prostředků, o které si uživatel zažádá, i když nejsou plně využity. Princip spočívá v přepínání paketů, takže více uživatelů využívá stejný datový kanál (TDMA kanál) a data jsou jim přenášena pouze po požádání v určitém časovém intervalu. Kapacita linky se tak přiděluje pouze těm účastníkům, kteří něco v danou chvíli přijímají nebo odesílají, čímž stoupá propustnost a to je vhodné zejména pro prohlížení internetových stránek, chatování, posílání mailů apod. Výhodou je i lepší tarifování, kdy lze platit jen za přenesená data nebo paušálně.

Další výhodou GPRS jsou podporované protokoly jako je IP, PPP, OSPIH, X.25 (využívá se pro platební terminály). X.25 byl však ze standardu dodatečně odebrán, i když je stále podporován přes IP a PPP.

### 2.2.1 Třídy zařízení GPRS

Zařízení, které podporují GPRS se rozdělují do tří tříd:

Class A – umožňuje používat zároveň GPRS a přitom přenášet i hlas, funkce DTM (Dual Transfer Mode). Tuto třídu musí podporovat jak mobilní telefon, tak i operátor.

Class B – umožňuje v jednu chvíli přenášet buď jen data nebo hovor, v souvislosti s aktuální podporou umožňuje při zapnutém GPRS přijmout hovor a naopak. Tato třída je povětšinou standardně podporována MS.

Class C – jedná se jen o datový provoz, nelze telefonovat. Jsou to např. přístrojové karty.

### 2.2.2 Kódová schémata

U GPRS existují čtyři kódová schémata, viz tab. 2.3. MS musí vždy všechna schémata podporovat, síť operátora GSM však nikoliv. Výběr kódového schématu se vybírá v závislosti na poměru signál/šum, aby byl možný co nejlepší přenos dat.

Tab. 2.3: Kódová schémata GPRS [5]

Kódové schéma	Rychlost [kbit/s]
CS-1	9,05
CS-2	13,4
CS-3	15,6
CS-4	21,4

### 2.2.3 Možnosti mobilních zařízení

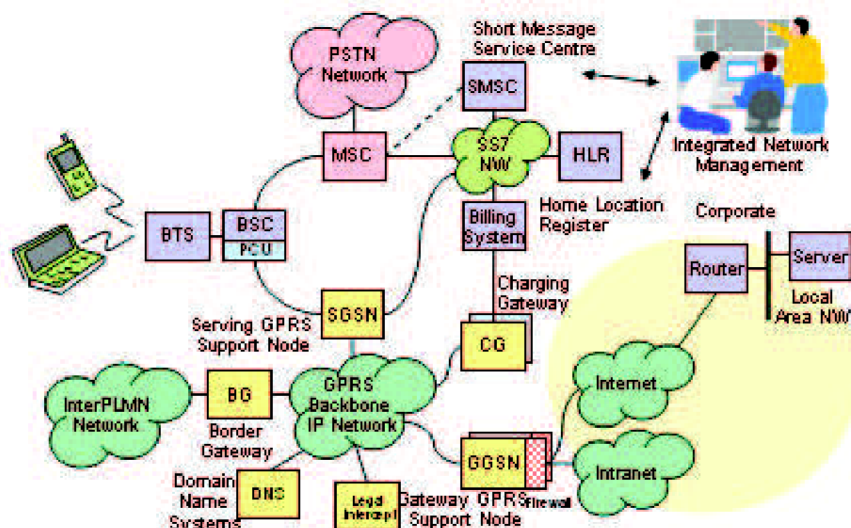
Podle možnosti, kolik timeslotů (TS) umí použít mobilní telefon pro downlink, uplink a kolik současně, se MS dělí do tříd, viz tab. 2.4, (Multislot Class). Běžné GSM/GPRS/EDGE terminály umí současně používat maximálně 5 timeslotů. Nejnovější terminály třídy 32 umí již používat 6 timeslotů, pokud toto podporuje i síť. Pro různé technologie (HSCSD, GPRS, EGPRS) může být třída Multislot Class rozdílná, nejběžnější je však třída 10. Nejpoužívanější jsou kombinace 4 + 1 nebo 3 + 2, ale mohou se měnit podle převládajícího toku dat.

Tab. 2.4: Přehled využití timeslotů podle podporované třídy MS [5]

Třída	Downlink TS	Uplink TS	Současně TS
1	1	1	2
2	2	1	3
3	2	2	3
4	3	1	4
5	2	2	4
6	3	2	4
7	3	3	4
8	4	1	5
9	3	2	5
10	4	2	5
11	4	3	5
12	4	4	5
32	5	3	6

### 2.2.4 Struktura GPRS

Přidání sítě GPRS do GSM byl nemalý zásah a stávající síť musela být upravena, protože byla dříve tvořena hlavně pro princip spínání okruhů (CSD). Schéma celé sítě GSM s GPRS je na obr. 2.5.



Obr. 2.5: Schéma sítě GSM s GPRS s rozepsanými anglickými zkratkami [6]

GPRS do sítě GSM vnáší další součásti, kterými jsou SGSN (Serving GPRS Support Node), GGSN (Gateway GPRS Support Node), BG (Border Gateway), CG (Charging Gateway) a LIG (Legal Interception Gateway).

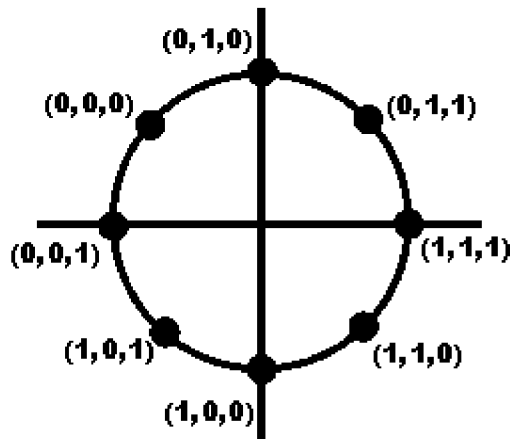
**SGSN** má podobnou funkci jako MSC, místo hlasu však přepíná datové pakety. Provádí směrování paketů a obsluhuje všechny uživatele, kteří jsou přihlášení do sítě a nachází se v dosahu. Dále pak provádí šifrování, autentizaci a kontrolu IMEI. Řídí logické kanály směrem k MS a spojuje HLR a MSC.

**GGSN** tvoří funkci rozhraní mezi GPRS a ostatními sítěmi pomocí protokolů IP a X.25. Kromě tohoto má za úkol směrovat pakety na správný uzel SGSN, funkce mobility managementu, firewall a konverze protokolů mezi GPRS a vnějšími sítěmi. GGSN a SGSN spolu komunikují skrz rozhraní GTP (GPRS Tunelling Protocol), které využívá transportní protokoly UDP a TCP.

Jednotka **BG** (Border Gateway) tvoří bránu pro GPRS a jiné operátory, kteří umožňují funkci roamingu. Jednotka **CG** (Charging Gateway) uchovává a vyhodnocuje zúčtovací data od SGSN a GGSN. Po celkovém zpracování je pak dále odesílá do celkového zúčtovacího systému sítě GSM. Jednotka **LIG** (Legal Interception Gateway) je jednotkou odposlechu, v některých státech je i povinná.

## 2.3 EDGE

S Release 98 byla představena i technologie EDGE (Enhanced Data rates for Global Evolution, Enhanced Data rates for GSM Evolution) jako 2,75G, která vylepšuje dosavadní přenosové technologie, kterými jsou GPRS a HSCSD. Hlavní vylepšení tkví ve změně modulace, kde byla stávající dvoustavová modulace GMSK (Gaussian Minimum Shift Keying) nahrazena osmistavovou fázovou modulací 8-PSK, viz obr. 2.6. Symbolová rychlost tak zůstala stejná (270,833 ksymbolu/s), ale bitová rychlost se zvětšila trojnásobně.



Obr. 2.6: Konstelační diagram 8-PSK

Na rozdíl od GPRS není primárně EDGE na každé BTS. Přítomnost signálu je indikována písmenem E u velikosti signálu na displeji MS nebo barevně.

Rozšíření EDGE zahrnuje dvě části, které navazují na předchozí technologie, a to EGPRS a ECSD.

## 2.4 EGPRS

EGPRS (Enhanced GPRS) vylepšuje GPRS a je s ní zpětně kompatibilní. Vylepšením je zejména změna modulace na 8-PSK, devět kódových a modulačních schémat (MCS-1 až MCS-9), viz tab. 2.5, přepracování vrstvy RLC/MAC, inkrementální redundance, větší délka okna, resegmentace a retransmise.

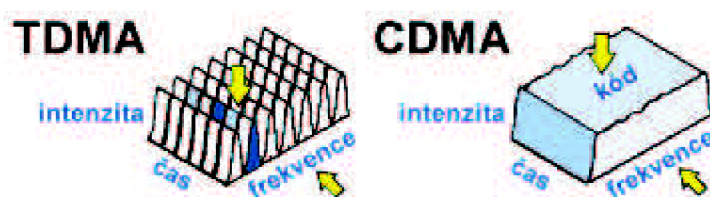
Tab. 2.5: Kódová schémata EGPRS [7]

Kódové schéma	Rychlost [kbit/s]	Modulace
MCS-1	8,80	GMSK
MCS-2	11,2	GMSK
MCS-3	14,8	GMSK
MCS-4	17,6	GMSK
MCS-5	22,4	8-PSK
MCS-6	29,6	8-PSK
MCS-7	44,8	8-PSK
MCS-8	54,4	8-PSK
MCS-9	59,2	8-PSK

Při porovnání GPRS a EGPRS vychází lépe EGPRS při stejné konfiguraci mobilního telefonu 4 + 1. GPRS přenáší data rychlostí 80 kbit/s při CS-4 a EGPRS rychlostí 200 kbit/s (reálně) pro download a 100 kbit/s pro upload u kódového schématu MCS-9.

## 2.5 CDMA

System CDMA (Code Division Multiple Access) přišel zhruba ve stejnou dobu jako systém GSM. Je ale rozšířen zejména v USA a Asii. Rozdíl mezi těmito dvěma technologiemi spočívá v práci s přenosovým kanálem. GSM využívá technologii TDMA (Time Division Multiple Access), kdy se jeden kanál vždy předává na krátký úsek jinému uživateli, viz obr. 2.7. CDMA si kanál v čase nepředává, ale všichni uživatelé na něj přistupují zároveň a odlišují se kódově. Podle kódu vybírají pro sebe důležitá data a zbytek je pro ně v podstatě šum. Toho je dosaženo pomocí tzv. konstruktivní interference speciálních kódů pro multiplexování, kde je použity PN sekvence pro zpětné kanály a Walshovy kódy pro předpřepřené kanály.



Obr. 2.7: Znárodnění přístupu TDMA a CDMA [8]

První síť založená na CDMA byla CDMAOne (IS-95A). Přenos dat u této verze byl 14,4 kbit/s, ale s vydáním IS-95B vzrostl až na 115 kbit/s, obdobně jako u GPRS u GSM.

Ve 3G s dalším vývojem vznikla rodina CDMA2000. První verze z této rodiny byla CDMA2000 1X, která oproti CDMAOne měla dvojnásobnou kapacitu pro přenos hlasu a maximální rychlost přenosu dat 307 kbit/s. Další verzí byla CDMA 1xEV-DO (Evolution Data Only) na kmitočtu 450 MHz. Ta nabídla max. download 2,4 Mbit/s a upload 153,6 kbit/s. Revize A této verze opět vylepšuje parametry a download vrostl na 3,1 Mbit/s a upload na 1,2 Mbit/s. Pro dosažení těchto rychlostí je použito kombinace CDMA a TDMA. Maximální rychlost přenosu je závislá na kvalitě signálu a pro download se většinou pohybuje v mezích od 38,4 kbit/s do 2,45 Mbit/s a pro upload od 9,6 kbit/s do 153,6 kbit/s. Ping tohoto signálu je kolem 250 ms.

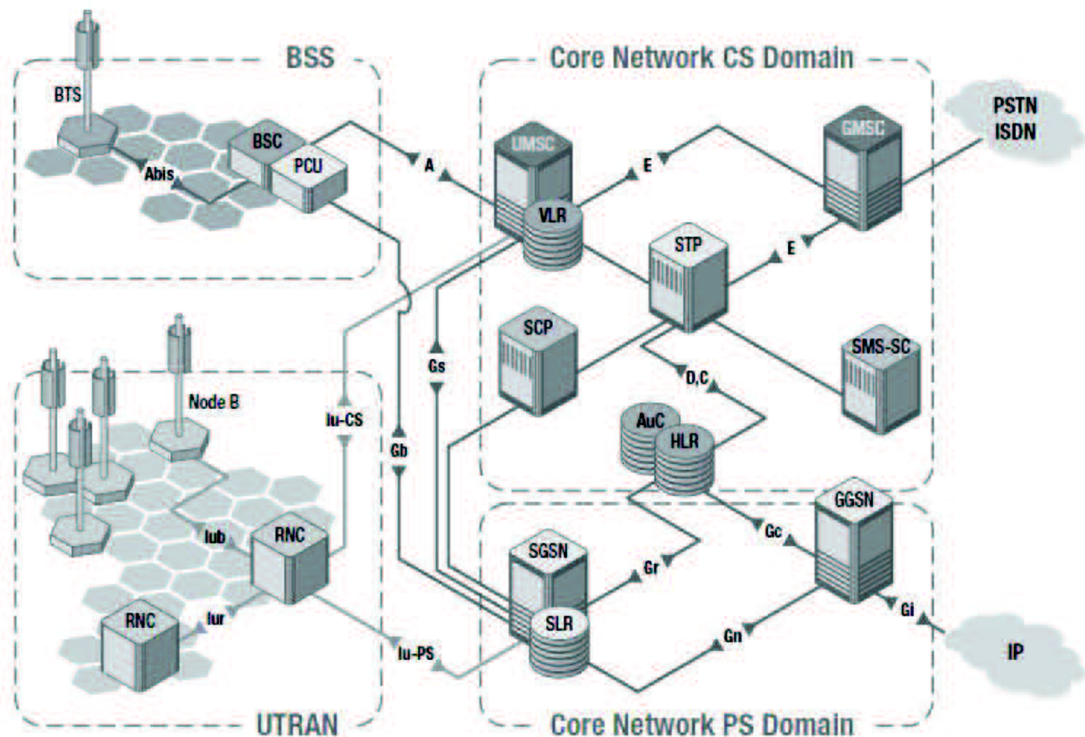
Dalšími rozšířeními jsou CDMA 1X a WCDMA. Tyto rozšíření jsou spíše zaměřeny na hlasové služby. Vychází to z principu, že se snaží zajistit konstantní datový tok směrem k terminálu, což má za následek snížení kapacity sítě. V tom je rozdíl oproti 1xEV-DO, protože tento systém umožňuje měnit každých několik ms datový tok k uživateli. Terminály přijímají signál z okolních BTS a vyhodnocují jeho úroveň, na základě toho se určí maximální rychlost, kterou může terminál přijímat data. Díky tomuto má uživatel optimální datový tok v závislosti na úrovni signálu. Jeden z dalších rozdílů je šířka spektra. U CDMA 1xEV-DO to je 1,25 MHz a u WCDMA je šířka kanálu 5 MHz.

## 2.6 UMTS

Standard třetí generace mobilních telefonů UMTS (Universal Mobile Telecommunication System) je koncipován jako nástupce systému GSM. Systém je standardizován organizací 3PP a je také evropským standardem splňující požadavky ITU IMT-2000. U UMTS je použit mnohonásobný přístup W-CDMA, který může být dále kombinován s TDMA a FDMA (Frequency Division Multiple Access).

### 2.6.1 Struktura sítě

Struktura sítě je patrná z obr. 2.8, kde je oproti GSM přidána část UTRAN (UMTS Terrestrial Radio Access Network) a některé bloky v jádře jsou pozměněny.



Obr. 2.8: Architektura UMTS s vyznačením pomocí zkratk (Release 99) [9]

Na obr. 2.8 chybí vyobrazení mobilní stanice UE (MS v systému GSM), bez které by pohled na síť nebyl kompletní.

Médium pro zajištění přenosu informací mezi uživatelskou stanicí a jádrem sítě se nazývá GERAN (podle standardu IMT-2000). GERAN není přesně definován a může ho tak tvořit jakýkoliv digitální komunikační standard jako je GSM, DECT, digitální televize nebo HiperLAN. Pro systém UMTS jsou určeny dva koncepty a to UTRAN a USRAN. Koncept USRAN (UMTS Satellite EAN) je satelitní rádiová síť a příliš se nevyužívá. Koncept UTRAN zajišťuje přístup k jádru sítě pomocí rádiové pozemní sítě.

UTRAN (GERAN) je tvořen ze základnových stanic nazvaných jako Node B (v GSM je to BTS) a ovladače rádiové sítě RNC (Radio Network Controller), což je v GSM označeno jako BSC.

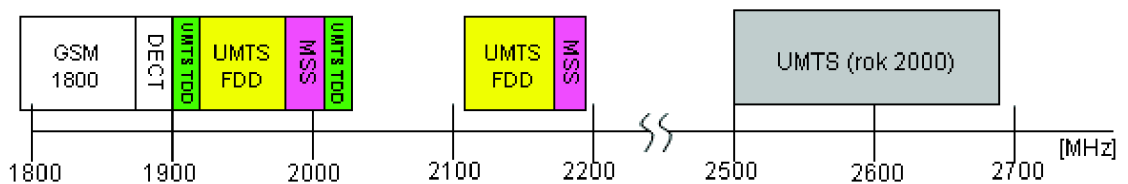
Dříve již bylo zmíněno jádro sítě, což je soubor služeb, které jsou poskytovány

UMTS síť a jeho rozhraní, pokud do nich uživatel přistoupí. Může to být např. telefonování do jiných sítí nebo připojení k internetu. Jádrem sítě se dělí na domény, které závisí na způsobu, jakým budou zákazníkovi poskytovány služby. Patří do nich tři základní, což je přepojování okruhů (CS), přepojování paketů a IP multimédia systém, tedy IMS doména. Kromě těchto základních k tomu patří i komponenty umožňující přepojování do jiných sítí, roaming, autorizaci, účtování atd.

Stejně jako u GSM jsou zde různá rozhraní, která propojují části sítě. Rozhraní mezi uživatelskou stanicí (UE, MS) a UTRAN je Uu (v GSM je to Um) a rozhraní mezi UTRAN a jádrem sítě je Iu. Rozhraní Iu závisí na rozdělení jádra, které může být buď paketově spínané nebo okruhově spínané. Pokud je paketově spínané, jedná se o rozhraní IuPS (Packet Switched), pokud je okruhově spínané, jedná se o IuCS (Circuit Switched).

### 2.6.2 Režimy oddělení uplinku a downlinku TDD a FDD

FDD (Frequency Division Duplex) je spojení mezi stanicí a telefonem na oddělených kmitočtech pro uplink a downlink. TDD (Time Division Duplex) je oddělení na základě timeslotů, tzv. „Ping-Pong“. Kmitočet však zůstává stejný. Rozložení kmitočtů UMTS ukazuje obr. 2.9.



Frekvenční rozsah	Šířka pásma	Použitá technologie
1900 - 1920 MHz	20 MHz	TD-CDMA
1920 - 1980 MHz	60 MHz	W-CDMA (vzestupný směr, účastník - síť)
1980 - 2010 MHz	30 MHz	Mobilní družicová služba
2010 - 2025 MHz	15 MHz	TD-CDMA
2110 - 2170 MHz	60 MHz	W-CDMA (sestupný směr, síť - účastník)
2170 - 2200 MHz	30 MHz	Mobilní družicová služba

Obr. 2.9: Rozložení kmitočtů UMTS [10]



## 2.7 WiMAX

WiMAX (Worldwide Interoperability for Microwave Access) je standart bezdrátové distribuce dat pro venkovní síť definován v normách IEEE 802.16, který je již systémem 4G. WiMAX pracuje v pásmech 2÷11 GHz a vyšším vysílacím výkonem a použitím směrových antén umožňuje velký dosah. Při přímé viditelnosti to může být až 50 km a několik km v městské zástavbě při použití spojů bez přímé viditelnosti (NLOS). K přednostem systému patří i velká kapacita připojení, kterou lze rozdělit mezi desítky klientů a každý přitom bude mít garantovanou stabilní přenosovou rychlost. Další vlastností WiMAXu je i podpora QoS.

## 2.8 Bluetooth

Ačkoliv technologie Bluetooth (IEEE 802.15.1) patří do kategorie sítí WPAN, byla do této kapitoly zařazena, protože většina komunikace, která probíhá mezi mobilním zařízením, připojeného k síti GSM, a počítačem probíhá právě přes rozhraní Bluetooth, viz hlavní zdroje pro tuto kapitolu [11] a [12].

### 2.8.1 Počátky a verze technologie Bluetooth

Vznik této technologie se traduje od roku 1994, kdy pracovníci firmy Ericsson na základě studie o nahrazení sériového připojení RS-232 k mobilnímu zařízení definovali první verzi tohoto rozhraní. V roce 1998 pak byla založena skupina SIG tvořena původně pěti firmami, s cílem vytvořit univerzální standard pro bezdrátovou komunikaci. První verze Bluetooth poté vyšla ve verzi 1.0 v červenci roku 1999. Jméno bylo odvozeno z anglického jména dánského krále Haralda Modrozuba (Haralda I Blaatanda), protože tento král vynikal svými diplomatickými schopnostmi a umožnil komunikaci znesvářených kmenů ke smíru, stejně jako technologie Bluetooth pomáhá ke komunikaci mezi různými zařízeními vybavenými touto technologií. Pro Bluetooth je i typické logo, viz obr. 2.10.



Obr. 2.10: Logo Bluetooth [13]

Technologie Bluetooth prodělala od počátku svého vývoje řadu změn. Verze 1.0a, vyšla v červenci roku 1999 a verze 1.0b následovala v prosinci téhož roku. Tyto první verze měly problémy s kompatibilitou, určováním role Master a Slave a implementací pikosítí. Z těchto důvodů vyšla v únoru roku 2001 verze 1.1, které byla přidána podpora pro nešifrované kanály a indikátor síly signálu. Technologie Bluetooth měla stále poměrně hodně nedostatků, a tak byla celkově přepracována. Výsledkem byla verze 1.2, která vyšla v listopadu roku 2003. Verze 1.2 byla rozšířena o možnost rychlého vytvoření připojení. Dále byla vybavena technologií AFHSS a povolena vylepšená kvalita hovoru v rámci připojení Extended SCO. V roce 2004 byla vydána verze 2.0, jež umožňuje připojení až rychlostí 2,2 Mbit/s pomocí EDR (2.0 + EDR). Další verze 2.1 + EDR, která vyšla v červenci 2007, umožňuje rychlejší párování zařízení a podporu pro Near Field Communications. Verze 3.0 vyšla v roce 2009 a využívá architekturu

MAC/PHY, která umožňuje fyzické propojení s Wi-Fi, a také protokolu 802.11 PAL. Přenosová rychlost se tak zvýšila zhruba desetinásobně a snížily se energetické nároky. Dalším vylepšením je EPC, které zmenšuje počet výpadků při přenosu dat u změny umístění zařízení. Poslední verze Bluetooth vyšla v červenci 2010. Jedná se o verzi 4.0 a svým charakterem nenahrazuje verzi předchozí. Důraz je kladen především na úsporu energie. Toho je docíleno umístěním řídicí jednotky přenosu přímo do čipu Bluetooth a také používáním velmi krátkých datových paketů. Přenosová rychlost je tak maximálně 1 Mbit/s (v některých zdrojích uváděna až 22 Mbit/s), té bude možné dosáhnout i při zhoršených podmínkách přenosu. Dalším vylepšením je šifrování přenosu AES-128.

## 2.8.2 Struktura a vlastnosti technologie a komunikace

Komunikační protokol v sobě zahrnuje všechny vrstvy, což v důsledku vytváří poměrně složitou technologii. Při realizaci zařízení jsou do něj však aplikovány jen nejnужnější části podle toho, jaký účel má dané zařízení plnit. Základem jsou vždy protokoly jádra a SDP, což zajišťuje komunikaci zařízení, které nepodporují stejné protokoly.

Bluetooth pracuje na kmitočtu 2,4 GHz, stejně jako Wi-Fi (z toho důvodu dochází mezi těmito technologiemi k značnému rušení), který se nachází v pásmu ISM. Používá principu FHSS s 79 (v Japonsku 32) nosnými s odstupem 1 MHz, mezi kterými přelazuje 1600krát za sekundu.

Pro komunikaci jsou definovány tři výkonové úrovně. Class 1 (100 mW (20 dBm), dosah zhruba 100 m), Class 2 (2,5 mW (4 dBm), dosah zhruba 10 m) a Class 3 (1 mW (0 dBm), dosah zhruba 1 m). Vzdálenost dosahu je uváděna ve volném prostoru. S překážkami roste chybovost, ale nevznikají výpadky spojení.

Komunikace mezi jednotlivými zařízeními se vždy řídí vztahem Master – Slave. Zařízení Master a až sedm zařízení Slave tak tvoří pikosíť (piconet). Na jednom prostoru může fungovat koexistence až deseti pikosítí, která se pak nazývá rozprostřená síť (scatternet). Odlišují se od sebe rozdílným schématem kmitočtových skoků na jednotlivé nosné kmitočty. Řízení pikosítě má na starosti zařízení Master. Podle něho se určuje na jakou nosnou bude v následujícím skoku přeskočeno, resp. které schéma bude použito. Schéma je určeno tak, že začátek pořadí frekvencí udává Master a poté se pokračuje v pseudonáhodném pořadí, které je určeno z adresy zařízení Mastera a stavu jeho CLK. Zařízení Master udává zařízením Slave časovou prodlevu 625  $\mu$ s. Po uplynutí této doby může teprve započít komunikace v pikosíti. Při komunikaci Master – Slave se používá metoda TDD, kdy Master začíná komunikaci v místě sudé časové prodlevy a zařízení Slave v místě liché časové prodlevy.

Jako modulace je použita GFSK (u verze 2.0 + EDR a výše je použita pi/4-DQPSK). Log. 1 jsou definovány jako kladné odchylky od nosného kmitočtu a log. 0 jako záporné. Rozdíl odchylek by měl být 115 kHz.

Zařízení se vzájemně identifikují pomocí 48-bitové adresy BD\_ADDR. Po přihlášení do sítě získají zařízení Slave tříbitovou adresu AM\_ADDR pro komunikaci v pikosíti. Kvůli krátké adrese může být k zařízení Master připojeno jen sedm zařízení. Při přihlašování do sítě se musí zařízení Slave synchronizovat. Prvně tedy musí najít všechny okolní zařízení, zjistit jejich názvy a stav CLK a poté naváže spojení.

Komunikace mezi dvěma zařízeními může probíhat buď synchronně nebo asynchronně. Synchronní spojovaná komunikace SCO vytváří symetrickou komunikaci

mezi zařízením Master a zařízením Slave. Zařízení Master posílá informace zařízením Slave v zadaných časových prodlevách, přičemž zařízení Slave může poslat informace v následující časové prodlevě.

Asynchronní nespojovaná komunikace ACL se využívá k vytvoření spojení mezi zařízením Master a jedním zařízením Slave, pokud přenosový kanál není rezervován SCO. Asynchronní komunikace odpovídá paketovému přenosu. Při komunikaci mohou komunikovat pouze dvě zařízení, ale může nastat výjimka, kdy pouze zařízení Master může odesílat zprávy na všechny zařízením slave pomocí broadcastu. Asynchronní připojení je určeno hlavně pro efektivnější přenos a z toho důvodu je kladen důraz na integritu dat a ne na případně zpoždění. Pakety, které nedojdou nebo jsou doručeny chybně jsou odesílány znovu.

Pro odesílání dat se využívá několika typů různých paketů. Pro připojení SCO a ACL jsou k dispozici obecné typy paketů s pakety s ID, NULL, POLL a FHS. Tyto typy paketů slouží ke správě účastnických stanic. Paket typu DM1 je základním prvkem spojení a využívá se v obou typech připojení. Pro přenos ladících informací během ACL přenosu je použit paket typu DH. Od typu DM se liší odstraněním dat FEC. Pakety typu HV jsou určeny pro přenos hlasu u SCO.

### 2.8.3 Bezpečnost

Při používání každé technologie je nutné dbát na bezpečnost přenosu dat. Technologie Bluetooth byla ve svých počátcích poměrně málo zabezpečena. To se postupným vývojem změnilo, ale stále platí několik bezpečnostních rizik, se kterými se uživatelé mohou setkat.

Délka použitého klíče pro komunikaci se vybírá podle pravidla kratšího klíče. Tedy délka souhlasí s nejmenší délkou klíče jednoho ze zařízení. Záleží pak tedy na délce klíče. Čím kratší klíč, tím lehčí je prolomit obranu.

Další riziko je kód PIN, který se zadává při párování zařízení. U některých zařízení je zadáván uživatelem, u některých je zadán přímo. Vzájemné ověřování kódů PIN není založeno na certifikátech, čímž hrozí riziko útoku typu Man in the middle.

Velkou bezpečnostní mezerou je absence zabezpečení u většiny aplikačních protokolů. Tento fakt je podstatný, neboť mnoho aplikačních protokolů může přistoupit přímo k adresářové struktuře daného zařízení.

Při výčtu těchto nedostatků je nutné dodat i některé typy útoků, kterýmž jsou např. Bluejacking, kde se jedná o posílání zpráv na jiná zařízení, čemuž se lze bránit vypnutím Bluetooth nebo zapnutím neviditelnosti pro ostatní zařízení, nebo již výše zmiňovaný Man in the middle, kde se jedná o odposlouchávání komunikace.

## 3 MĚŘENÍ PARAMETRŮ SÍTĚ GSM

Pro správnou funkčnost sítě je potřebné měření parametrů sítě a jejich vyhodnocování. To zajišťuje dostatečnou kvalitu sítě, kterou provozovatelé mohou nabídnout svým zákazníkům. Aby bylo dosaženo co nejlepších výsledků, provádí se nepřetržité monitorování kvality sítě a služeb – QoS (Quality of Service), [14]. Mezi nejdůležitější parametry patří, [14]:

- NA (Network Availability) [%] – dostupnost sítě (služby),
- CSR (Call-Success Rate) [%] – procento úspěšně ukončených hovorů,
- CSSR (Call-Setup Success Rate) [%] – procento úspěšně navázaných hovorů,
- HSR (Handover Success Rate) [%] – úspěšnost předání hovoru (handover),
- Speech (voice) quality [MOS, PACE, Q..] – soubor hodnot reprezentující výsledek objektivního (automatického) měření kvality řeči v síti,
- Set-up time [s] – průměrný čas sestavení hovoru (od vytočení po spojení),
- RTD (Round Trip Delay) [ms] – čas odezvy IP sítě na PING,
- GPRS throughput [kbps] – průměrná hodnota propustnosti sítě pro PD (Packet Data).

### 3.1 Parametry sítě GSM a měřící metody

Vzhledem k tomu, že okolní vlivy na mobilní síť jsou značné a také poměrně proměnlivé (příkladem mohou být změny v infrastruktuře nebo jiné měnící se vlivy v prostředí), je potřebné stále sledovat parametry sítě (QoS). To lze dosáhnout několika způsoby, kterými jsou např. OMC (Operating and Maintenance Centre), trasování hovorů (Call trace) nebo měření za jízdy (Drive test). Kromě testování je potřebné provádět údržbu sítě a servis zařízení, viz zdroj informací pro tuto podkapitolu [14].

Další cestou, jak dosáhnout přehledu o síti je simulování sítě se známými parametry, které jsou nastaveny dle požadavků nebo změřeny. Simulace probíhá poměrně přesně, ale nemůže plně nahradit přímé výsledky měření, protože nemusí předpokládat nebo zahrnovat některé nepředpokládané vlivy. Pro popis tohoto způsobu získávání parametrů o síti zde ale není prostor.

#### 3.1.1 OMC

Pomocí registrů událostí, dostupných z OMC, je možné získávat přehled o kvalitě a funkčnosti sítě. Na terminály dohledového střediska správy sítě, které provádí nepřetržitou kontrolu, přicházejí informace o jednotlivých stavech prvků a části sítě. Potřebné údaje se pak získávají z definovaných registrů a z dat se poté vytvářejí statistiky, jako jsou např. vytížení, provoz, úspěšnost sestavení hovoru, ale také parametry jednotlivých buněk nebo i sítě. Pomocí OMC lze do sítě i zasahovat a měnit parametry nastavení nebo zapínat a vypínat prvky v případě poruchy. Pomocí OMC lze tedy kontrolovat celou síť a získávat z ní údaje pro statistiky,

ale není možno získávat hodnocení kvality řeči.

Pomocí získaných dat se mohou sestavovat statistiky jako CSR, CSSR, HSR (daný počtem pokusů o handover a počtem provedených handoverů), dále pak statistika vytížení buňky (Load Intensity) nebo statistika blokování pokusů o sestavení hovorů (Blocked Calls), statistiky datových přenosů a služeb, vytížení ústředí, linek apod.

### 3.1.2 Call trace

Trasování hovorů a událostí v síti CT (Call Trace) umožňuje zpětně analyzovat události v síti, neboť veškeré události v síti jsou ukládány do databáze a pomocí vzdáleného přístupu lze jednotlivé soubory zpětně načíst a filtrovat pro zjištění potřebných údajů. Toto je možné díky standardizaci jednotlivých rozhraní sítě mezi jejími prvky (BTS, BSC, MSC, HLR apod.), které jsou napojeny do centrálního protocol analyzoru. Nespornou výhodou je tedy zpětná analýza událostí, např. zjištěných poruch, ale velká nevýhoda je nemožnost stálého analyzování sítě, protože vzhledem k jejímu rozsahu by vznikaly velké nároky na úložná a zpracující zařízení.

### 3.1.3 Drive test

Drive test se od dosavadních možností získávání dat liší, protože získávání dat probíhá prostřednictvím vozidla, které provádí datové přenosy a uskutečňuje hovory a je vybaveno měřicím systémem a GPS lokalizací. Měření tedy probíhá v reálném čase, ale je poměrně časově a finančně náročné. Navíc výsledky měření jsou určeny z jedné malé oblasti a neposkytují tak větší přehled o síti. K výhodám ale patří možnost srovnání sítě s jinými provozovateli.

Testování hovorů a přenosu dat může probíhat buď samostatně, nebo v součinnosti se stacionární jednotkou. Ve spolupráci s ní je pak možné získávat data z downlinku i uplinku. Od ostatních testů se liší v možnosti testovat kvalitu řeči. Pomocí měřicího systému je vyslán nahraný záznam, který se pak zpětně porovnává a pomocí něhož se získává přehled o chybách při přenosu.

Aby bylo možné testování provádět nepřetržitě a také na těžko dostupných místech, jako je železnice apod., jsou mobilní měřicí stanice (unit, black box) umístěny do vozidel MHD (městské hromadné dopravy), drážních vozidel železnice, vozidel taxi služby, vozidel autodopravy apod., které jsou kontrolovány centrální jednotkou (fleet manager). Jednotky realizují dávky testovacích datových a hlasových přenosů na základě přednastaveného programu. Mezi jednotlivými dávkami jsou získaná data odesílána centrále, která i jednotky ovládá. Výhodou je, že jde o samostatný zautomatizovaný test, ale nevýhodou je neznalost předchozí trasy.

### 3.1.4 Měření kvality řeči

Předmětem této práce není zaměření na posuzování kvality řeči, ale je zaměřena na vyhodnocování datových služeb. Přesto je zde uveden teoretický základ pro úplnost části měřicích metod.

S nástupem digitálních technologií rostl i požadavek na měření kvality přenášeného hlasu k uživateli. Jedná se o vyhodnocování vlivů částí sítě nebo i sítě jako celku. Účinek telekomunikačních zařízení lze nalézt pod názvem transmission quality nebo také speech quality (voice quality) u úzkopásmových sítí.

Měření kvality řeči je možné provádět dvěma způsoby a to buď pomocí subjektivní metody nebo objektivní. Subjektivní metoda je poměrně hodně časově a finančně náročná, avšak pracuje na jednoduchém principu. Z toho důvody byly vyvíjeny objektivní metody, které by umožnily všeobecné srovnání a urychlily vyhodnocování. Proto byl vyvinut způsob objektivního měření kvality řeči na základě subjektivního měřítka.

U subjektivních metod se hodnotí srozumitelnost telefonního hovoru posluchači. Posluchači mohou být různých skupin a jejich výběr je určen podle daných kritérií. Subjektivní metody můžeme rozlišit na dvě základní a to na konverzační, kdy spolu komunikují dva lidé v oddělených zvukotěsných místnostech a podle kvality poslechu určují srozumitelnost dle přesně dané stupnice, a poslechové, kdy několik posluchačů poslouchá dané nahrávky. Struktura a způsob přehrávání je dán použitou poslechovou metodou např. ACR (Absolute Category Rating), DCR (Degradation Category Rating) a CCR (Comparison Category Rating). Metoda ACR je doporučována ITU-T a proto je i nejméně rozšířena.

Pro vyhodnocování kvality přenášeného zvuku u objektivního modelu bylo vyvinuto několik algoritmů, z nichž se prosadil PESQ (Perceptual Evaluation of Speech Quality). Princip je založen na porovnání původního referenčního vzorku s degradovaným signálem (zkresleným průchodem skrz komunikační systém). Výsledkem je predikce výsledné úrovně kvality, kterou by byl poslech ohodnocen při subjektivním testu ACR.

## 3.2 Měřicí systém QVoice

System QVoice od firmy Ascom patří mezi nástroje pro vyhodnocování mobilní sítě QoS. Firma jej nabízí v několika provedeních, která se od sebe liší velikostí a rozsahem možností. Pomocí systému QVoice je možné testovat hlasové i datové služby a to podle předem definovaných scénářů nebo podle vlastních, viz zdroj tuto podkapitolu [15].

### 3.2.1 QVoice Smart

Nejmenší zástupce systému QVoice je skládá z mobilního telefonu, který je připojen přes datový kabel k cestovnímu počítači (notebooku) a softwaru, který je nainstalován na počítači, viz obr. 3.1. Jedná se tedy o cestovní zařízení, které umožňuje získávat požadované informace a vytvářet pomocí nich vlastní statistiky.



Obr. 3.1: QVoice Smart

### 3.2.2 QVoice Companion

Další zástupcem systému QVoice je „cestovní batoh“, ve kterém jsou ukryty všechny prvky pro měření sítě. Batoh obsahuje i baterie, takže systém může být stále v pohybu, ať už při jízdě automobilem nebo pěšky. U této již není potřeba počítač, protože informace o měření a nastavování je prováděno pomocí tabletu, viz obr. 3.2.



Obr. 3.2: QVoice Companion

### 3.2.3 QVoice Compact

Posledním mezi menšími zástupci je zařízení, které kombinuje dvě předešlé. Jedná se o cestovní kufřík, který obsahuje několik měřících modulů a je možné se k němu připojit pomocí cestovního počítače, viz obr. 3.3.



Obr. 3.3: QVoice Compact

### 3.2.4 QVoice Symphony

Další skupinou v systému QVoice jsou větší jednotky, které obsahují několik výměnných modulů, které lze dle potřeby měnit, viz obr. 3.4. Podporují 1 až 24 kanálů a to jak modemy nebo mobilní telefony. Jsou podporovány generace od 2G do 4G. Pro jednoduchost je možné využít jednodušší verze, které obsahují měření kvality hlasu, obrazu, data o provozu stanice (selhání spojení, handovery), SMS, MMS, FTP, HTTP atd., nebo využít podrobnější verze, která obsahuje i sílu signálu, rušení v okolí, kvalitu signálu apod.



Obr. 3.4: QVoice Symphony

### 3.2.5 QVoice Symphony Multi

Poslední verze systému QVoice, viz obr. 3.5, nabízí kombinaci předchozích vlastností systému QVoice Symphony, ale také možnost konfigurovat vlastní test a také provádět testy z pohledu účastníka a netechnického uživatele. U tohoto nového systému je také rozšířena podpora mobilních telefonů a nových technologií a také možnost plně automatizovatelnosti podle přednastaveného scénáře.



Obr. 3.5: QVoice Symphony Multi

### 3.2.6 QVoice Presentation

Pro ovládání zařízení a vyhodnocování dat je určen program QVoice Presentation. Jeho struktura je modulární a skládá se z prvků:

- QVP Map – zobrazení bodů měření, zobrazení buněk, překrývání, analýzu apod.,
- QVP Network Analyser – umožňuje analyzování ziskných dat a tvorbu grafů a tabulek,
- QVP TopView – založen na Google Earth, umožňuje zobrazovat síť ve 3D,
- QVP Report – umožňuje tvořit a vyhodnocovat benchmarky,
- QVP Expert – umožňuje samostatně provádět diagnostiku a vyhledávat problémy.

## 3.3 Měřicí systém Tems

Systém Tems byl vyvíjen firmou Ericsson a poté převzat firmou Ascom. V nynější době se tedy jedná o jednodušší měřicí systém, než je systém QVoice, ale v mnoho směrech ho doplňuje. Stejně jako systém QVoice je rozdělen do několika variant, viz zdroj pro



tuto podkapitolu [15].

### **3.3.1 TEMS Pocket**

TEMS Pocket je samostatný mobilní telefon určený k měření a zjišťování parametrů o síti. Jedná se tedy o momentálně nejmenší nástroj pro měření na trhu. Podporuje GSM/GPRS, EDGE, WCDMA, HSPA, CDMA a EV-DO. Změřené údaje je možno zobrazit na barevném displeji nebo uložit pro pozdější analýzu. V nabídce jsou čtyři varianty a to Professional, Classic, Premium a Remote.

### **3.3.2 TEMS Investigation**

Další variantou je nástroj pro měření sítě na mobilním počítači s připojeným zařízením. Vyhledává hlavně potíže se sítí, ale dovede i síť testovat. Slouží ke sběru dat v reálném čase a jejich následném zpracování. Je zde i přidána podpora HSPA+, LTE, WiMAX, iDEN a TD-SCDMA.

### **3.3.3 TEMS Automatic**

Tato varianta je určena pro využití jako black boxy v prostředcích MHD, taxislužeb apod. a lze je zařadit do kategorie automatizovaných Drive testů. Jedná se tedy o sondy do běžného provozu, které se liší provedením podle toho, pro jaký účel jsou určeny.

### **3.3.4 TEMS Software**

Kromě výše popsaných hardwarových nástrojů je komplet TEMS doplněn i o softwarové řešení, které se zabývá plánováním a analýzou sítě nebo zpracováváním naměřených údajů a jejich rozlišováním. Tyto nástroje jsou TEMS Box Office, TEMS Discovery a TEMS Visualization.

## 4 MĚŘENÍ PARAMETRŮ SÍTĚ GSM DLE POŽADAVKŮ UŽIVATELE

V této kapitole je zrekapitulován postup řešení a nastíněno výsledné řešení návrhu programu pro měření a vyhodnocování sítě GSM. Výsledné řešení je uvedeno v následující kapitole. Základem je zpracování požadavků, které jsou na program kladeny a jejich řešení a také možnosti, kterými lze těmto požadavkům vyhovět.

### 4.1 Požadavky na řešení

Aby požadovaný software byl vytvořen jako program schopný zhodnocovat pohled běžného netechnického uživatele, bylo stanoveno několik základních požadavků, které by měly určit rozdíl mezi změřenými statickými parametry a parametry důležité z pohledu uživatele.

Jednotlivé požadavky byly postupně zpracovávány a rozvrstveny do verzí, viz následující kapitola. Základními však jsou:

- schopnost komunikovat se třemi a více mobilními telefony zároveň (nejlépe pomocí Bluetooth) a měřit a vyhodnocovat tak zároveň jeden požadavek v sítích různých operátorů,
- možnost vytvářet dlouhodobý plán práce programu, aby tak nebyl závislý na nutnosti přímého ovládání, ale ve stanovený čas by prováděl stanovené operace a data o nich by ukládal a vyhodnocoval,
- bude provádět vyhodnocování datových služeb, nikoliv hlasových,
- vyhodnocování a testy budou prováděny z pohledu uživatelského, takže se nejedná o technologické posouzení kvality parametrů sítě.

Dále by bylo vhodné, kdyby program umožňoval:

- měření dat za pohybu měřicí stanice a tak i MS,
- vlastní přepínání mezi jednotlivými BTS v dosahu řízené mobilním telefonem,
- výběr požadované technologie pro přenos dat,
- zpětnou vazbu z mobilního telefonu, aby bylo možno sledovat jeho chování.

Mezi počítačnými podmínkami byly i zahrnuty testy, které by měl program obsahovat a které pak bude následně vyhodnocovat:

- test PING s proměnlivou délkou velikosti paketů, počtu testování a času,
- porovnání různých technologií mezi sebou (EDGE/GPRS),
- úspěšnost připojení k datovým službám,
- kvalita datových služeb v závislosti na signálu,

- hodnocení přehrávání videa a přehrávání hudby na mobilním telefonu z internetových stránek,
- možnost sledování velikosti streamu, např. z [www.youtube.com](http://www.youtube.com),
- stabilita internetových stránek a také úspěšnost práce s nimi při nízkém signálu a s tím také sledování velikosti signálu,
- hlasová a video komunikace pomocí protokolu SIP, komunikace např. několika klientů ICQ či Skype.

Řešení jednotlivých požadavků je uvedeno v následující kapitole v rozboru funkce programu a jeho testů. Požadavky však nebyly řešeny všechny, protože jejich řešení je závislé na aktuální verzi vyvíjeného programu.

## 4.2 Postup v řešení

V průběhu řešení projektu se objevilo několik cest, které nebylo možno realizovat nebo jen velice problematicky.

### 4.2.1 Prostředí Matlab

Pro řešení projektu bylo původně vybráno prostředí Matlab, které umožňuje usnadnění používání v již vytvořených Toolboxech a také umožňuje zkombinování více programovacích jazyků, jako je např. C nebo C++/C#. Tím by bylo možné do Matlabu vkládat vlastní zdrojové kódy, které by přesněji určovaly práci jednotlivých částí při nevyhovujících Toolboxech nebo by lépe definovaly požadované funkce.

Tyto výhody Matlabu však sebou nesou i mnoho nevýhod. Toolboxy, které Matlab obsahuje, sice mají mnoho prvků, které práci usnadní, ale tyto prvky jsou rozesety mezi několika součástmi Matlabu a jejich kombinace značně a zbytečně zvětšuje program a také komplikuje celkovou práci. Mimo toto je pak problém ve slučování více programovacích jazyků, protože vložené moduly spolu nefungují korektně.

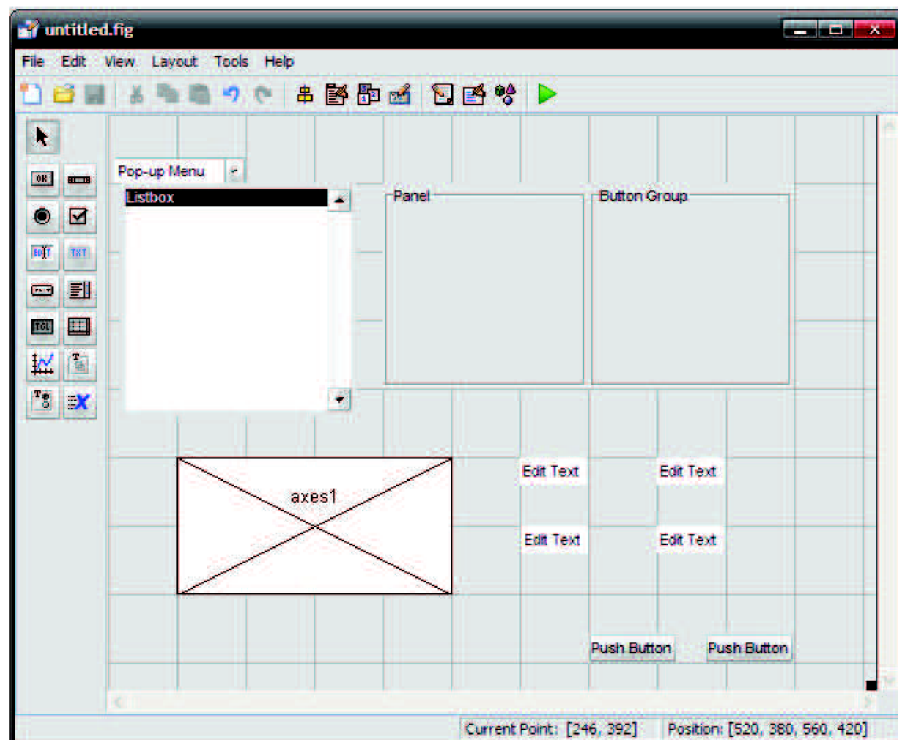
Matlab dále umožňuje využít vývojové prostředí GUI, viz obr. 2.1. Jednotlivé ovládací prvky se vloží do předpřipraveného pole a zdrojový kód k ovládacím prvkům se napíše do jednotného M-filu. To při programu s více ovládacími prvky působí poměrně nepřehledně.

Kromě ovládacích prvků lze využít prvky ActiveX. To jsou prvky dodané od ostatních producentů software, jako je např. Microsoft. Pro využití v projektu by byl vhodný např. prvek ActiveX Skype, který by umožňoval vizuální a hlasovou komunikaci přes internet a tak např. následné vyhodnocení parametrů obrazu, přičemž by se analyzoval vyslaný obraz, přijatý a výsledky by se porovnály. Prvky ActiveX mají poměrně značnou nevýhodu, tím že se jedná o prvky třetích stran, protože jejich programové vstupy a výstupy jsou přesně dány a není možné do nich zasahovat.

Další omezení Matlabu, které se vyskytlo, jsou poměrně malé možnosti práce s internetem. Matlab umožňuje načtení internetové stránky pomocí příkazu *urlread*, uložení internetové stránky pomocí příkazu *urlwrite*, dále pak posílání e-mailu pomocí příkazu *sendmail* a jako poslední příkazy pro komunikaci skrz protokol FTP, které je možno najít v FTP class. Zobrazování internetových stránek navíc probíhá ve vlastním integrovaném prohlížeči.

Poměrně problematická byla i komunikace s mobilním telefonem skrz sériový port a ovládání mobilního telefonu pomocí AT příkazů. Matlab dovedl vytvořit sériový kanál a odeslat příkaz, ale problém mu již činilo zachycení odpovědi. Důvod je možná způsoben tím, že Matlab hledal odpověď ve svém prostředí a tak zobrazoval hodnoty, které byly odesílány do sériového portu a ne ty, které byly přijímány. Při vzniku chyby programu bylo nutné telefon znovu připojit a spárovat. Po této operaci byl k přístroji vždy přidělen nový port COM a program se musel manuálně přenastavit. Pokud by takový program pracoval v Matlabu samostatně, byl by pak vždy vyžadován zásah obsluhy, viz ukázka zdrojového kódu:

```
s = serial('COMx','BaudRate',921600,'DataBits',8); /*nastavení
                                                    kanálu pomocí sériového portu na rychlosti
                                                    921600Bd (Bluetooth), poslání dat o délce
                                                    8b, x značí číslo portu */
fopen(s); /* otevření datového kanálu */
s /* výpis stavu portu */
fprintf(s,'xyz'); /* odeslání AT příkazu (xyz) */
idn = fscanf(s); /* přečtení odpovědi zařízení uložení do
proměnné idn */
fclose(s); /* zavření datového kanálu */
idn /* výpis odpovědi */
s /* výpis stavu portu */
```



Obr. 4.1: Ukázka grafického prostředí GUI

## 4.2.2 Výběr mobilního telefonu

Prostředek komunikace vytvářeného programu a síť GSM je mobilní stanice. Předpoklad je, že mobilní stanice bude připojena k počítači, na kterém pracuje vytvářený program, přes rozhraní Bluetooth. Bude tak vytvořen sériový kanál na portu

COM, skrz který se budou posílat do mobilní stanice AT příkazy a také očekávat na ně odpovědi a mimo to stahovat případná data. Kromě připojení mobilní stanice přes port COM bude mobilní stanice sloužit i jako modem pro komunikaci s internetem.

Při takto předpokládané konfiguraci se zde vyskytuje několik omezení, která lze předpokládat. Jedním z nich je možnost, kdy na mobilní stanici začne pracovat vlastní program (vycházející z ovládacího software mobilní stanice předinstalovaný výrobcem zařízení) a nebude tak možno ovládat mobilní stanici pomocí AT příkazů. Další omezení vyplývá z faktu, že ovládací software mobilní stanice předinstalovaný výrobcem není možné měnit nebo jinak do něj zasahovat a v důsledku toho není možné realizovat některé požadavky na vytvářený program. Jedná se hlavně o hodnoty přístupu k internetu nebo ke službám. Přesné hodnoty tedy nepůjdou ze systému určit a bude nutné je získávat jinými cestami, např. získávat hodnoty již ve vytvářeném programu. Pak zde ale vzniká poměrně velké zkreslení, protože data budou zpracovávána počítačem, poté budou poslána přes Bluetooth do mobilní stanice, kde je poté bude teprve mobilní stanice zpracovávat. Po zpracování bude očekávána odpověď, která poputuje stejnou cestou nazpět. Velikost zkreslení tu snad jen snižuje fakt, že všechny připojené mobilní stanice budou mít stejné podmínky, bude se lišit jen síť operátora, ke které budou připojeny (pokud bude více mobilních stanic použito a získané hodnoty se budou porovnávat). Jako další omezení zde vystupuje nemožnost ovlivňovat provádění handoverů mobilní stanice nebo určování používané technologie.

Pro dosažení požadovaných cílů je potřebné vybrat i vhodnou mobilní stanici, která bude komunikaci realizovat. Kritériem pro výběr může být podpora potřebných technologií a také omezení počtu AT příkazů, které pro všechny mobilní stanice nejsou stejné a krom základních se liší.

Možným výrobcem mobilních stanic, který by přicházel v úvahu, je Nokia. Modely této značky mají poměrně rozsáhlou podporu nových technologií a dobrou zákaznickou podporu produktů. Nokia na svých internetových stránkách přímo nenabízí druh technické podpory, který by byl vhodný, plně však spolupracuje a řeší případné dotazy, případně je možné potřebné materiály vyhledat. Modely firmy Nokia, které by přicházely v úvahu, jsou např. Nokia 2700 Classic nebo Nokia E52.

Podobně je na tom výrobce Sony Ericsson. Tato společnost na svých internetových stránkách nabízí přímo podporu pro vývoj aplikací a dokumenty po zaregistrování, dále pak i možnost přímé komunikace mezi vývojáři či techniky Sony Ericssonu nebo rozsáhlé FAQ. Modely, které by přicházely v úvahu, např. mohou být SE J108i Cedar nebo SE J10i2 Elm.

Mezi výrobce, kteří mají podporu spíše omezenou nebo neodpovídají na případné dotazy, patří Samsung. O výrobcích této značky se dá uvažovat, protože splňují technické předpoklady, ale omezenost podpory a materiálů spíše komplikuje práci. Modely Samsung Wave nebo Samsung Omnia II by byly vhodnou alternativou k předchozím.

### **4.2.3 Java aplikace na MS**

Vytvoření Java aplikace pro mobilní stanice by mohlo odstranit některé obtíže a omezení, která vznikají, pokud není možný přístup do softwarové a hardwarové konfigurace mobilní stanice.

Jedná se tedy o Java aplikaci, která by byla schopná vynahradit všechny nedostatky

mobilní stanice. Doplnovala by tak vytvářený software a rozšiřovala by možnosti využití, případně by byla schopná i pracovat samostatně. Nevýhody, které z toho plynou, jsou větší nároky na hardware mobilní stanice, protože by byl požadován dostačující výkon pro chod aplikace, a pak také nutnost napsat aplikaci v dalším programovacím jazyku s využitím platformy J2ME. Krom tohoto je zde i otázka, do jaké míry bude vůbec možné dosáhnout stanovených cílů pomocí aplikace na mobilní stanici. Vzhledem k časové náročnosti není tato možnost příliš realizovatelná. K možnostem, které do této kategorie i spadají je upravení a doplnění stávajícího operačního systému pro mobilní stanice, který je pod licencí GNU.

#### 4.2.4 GSM moduly

Další možnost, jak by bylo možno obejít softwarovou i hardwarovou uzavřenost vstupu do mobilního telefonu by bylo využití GSM hardwarových modulů.

GSM moduly by umožňovaly přesnější plnění pokynů dle vlastního programu a také by umožnily přímo kontrolovat data, která by bylo možno vyčítat. GSM moduly jsou primárně určeny pro aplikace jako je střežení a zabezpečení objektů, dálkové ovládání a monitorování strojů, případně telemetrie.

Jejich ceny a možnosti se různí. Mohou se pohybovat zhruba za hranicí 1000 Kč, kde se jedná o moduly GSM modul TM1(2), viz obr. 4.2. Tyto moduly obsahují funkce: TCP/IP Stack, uložení a čtení flash paměti, GPIO, CSD, SMS (text/data), Dual-Band GSM 900/1800 nebo 850/1900 MHz, sběrnici I2C a SPI, AT příkazy, GPRS třídy 10, případně další funkce lišící se dle typu. Výrobce těchto modulů nabízí i zvláštní doplňky na žádost zákazníka.



Obr. 4.2: GSM modul TM2 [16]

V cenové kategorii 6000 Kč a více se objevují GSM moduly, které nabízejí větší možnosti v oblasti mobilní komunikace, jako je kombinace použitých schémat u GPRS apod. Jejich vyšší cena je ale spíše způsobena pokrokem miniaturizace a rychlosti zpracovávání informací. Příkladem je GPRS/GSM modem Wismo Quick, viz obr. 4.3.



Obr. 4.3: GPRS/GSM modem Wismo Quick [16]

Uvedené GSM moduly by byly vhodné např. při vytváření kompletního hardwaru pro měření, neboť nabízejí hardwarové řešení, které je možné doplnit o vlastní softwarové. Bohužel realizace takového zařízení by byla poměrně náročná a svými vlastnostmi by se spíše začínal přibližovat komerčním měřicím systémům jako je např. QVoice, byť by další podstata řešení a zpracovávání dat byla jiná.

### 4.3 Zvolené řešení

Po zhodnocení předchozích zkušeností s řešením práce bylo odstoupeno od použití prostředí Matlab, aby v něm byl vytvořen požadovaný program, a místo něj byl vybrán programovací jazyk C/C++ a vývojové prostředí Borland C++ Builder 6.0.

#### 4.3.1 Postup a předpoklady

Aby dále nedošlo ke komplikacím při tvorbě uživatelského programu, byla nejprve vyzkoušena možnost propojení mobilní stanice s počítačem přes rozhraní Bluetooth. Propojení bude realizováno přes sériový port, skrz který se budou posílat AT příkazy a bude na ně i očekávána odpověď. Dále by pak bylo vhodné vyzkoušet připojení k internetu přes mobilní stanici s tím, že požadavky bude vysílat tvořený program.

Po vyzkoušení funkce jedné mobilní stanice bude vhodné připojit i další a vyzkoušet jejich společnou funkci. Zde by se mohl vyskytnout problém ve spolupráci operačního systému s mobilními stanicemi, protože systém Microsoft Windows XP se pro přístup k internetu připojuje jen k jednomu modemu. Pokud by bylo potřebné vysílat stejná data do sítí různých operátorů a následně tak vyhodnocovat jejich odlišnosti, bude potřebné, aby všechny mobilní stanice fungovaly jako modemy a to nezávisle na sobě.

Za předpokladu, že bude vyřešena komunikace více zařízení s počítačem zároveň, bude možné do programu vkládat nadstavby, které by sloužily k vyhodnocování získaných parametrů. Program by pak mohl být řešený modulárně, kdy bude vytvořen komunikační základ s ovládním a k tomuto budou přidávány knihovny umožňující měření a vyhodnocování dle vlastních definic.

Jednotlivé části by mohly být:

- kostra programu, na kterou budou moduly navazovat a pracovat pomocí ní,
- modul pro komunikaci s MS přes Bluetooth nebo jiné rozhraní (USB, RS232), tato část by i spíše mohla být přímo součástí základní kostry,
- modul pro plánování akcí,
- modul pro PING a jeho vyhodnocení,

- modul pro určení cíle k přehrání videa nebo muziky a vyhodnocení,
- modul pro komunikaci přes internet,
- modul pro nastavení parametrů měření signálu, vyhledání stanic a určení měřených stanic, nastavení základních parametrů pro chod programu (tento modul může být samostatný nebo může být později sloučen s kostrou),
- modul pro vyhodnocení, případně tvorbu jednotlivých testů, které by pak byly přímo definované uživatelem,
- modul pro export, zpracování výsledků měření a také průběhu měření.

Realizace prvotního návrhu jednotlivých modulů je uvedena v následující kapitole, včetně změn, které postupně prodělaly při vývoji jednotlivých verzí.

### 4.3.2 Ovládací software

Další fází projektu je samotný program, skrz který probíhá ovládání mobilní stanice, a který také vyhodnocuje získaná data. Pro vytvoření programu bylo použito prostředí Borland C++ Builder 6.

Pro vyzkoušení komunikace byla nejprve vytvořena aplikace posílající AT příkazy mobilní stanici a také získávající případnou odpověď na příkaz. Pro toto je potřebné vytvořit v aplikaci sériový kanál, skrz který budou posílána data, a také funkce, které budou obstarávat zápis a čtení dat.

Sériový kanál lze vytvořit jednoduchou deklarací funkce *fopen()*, kdy pro komunikaci je použit port COM. Podle toho, na jaký port je připojeno zařízení se odvíjí i číslo portu, viz následující syntaxe.

```
FILE *comport;          /* deklarování portu COM */
comport = fopen(port, "wr"); /*otevření      portu      pro
                             zápis a čtení,   proměnná   port
                             udává o jaký port se jedná, např.
                             COM5 */
```

K posílání řetězce dat (AT příkazu) slouží funkce *int sprintf(char \*retezec, char \*format, arg1, arg2, ...)*, kde zformátované argumenty podle řetězce format jsou umístěny do proměnné retezec. K vyčítání řetězce dat slouží funkce *int sscanf(char \*retezec, char \*format, arg1, arg2, ...)*, kde retezec podle formátu v řetězci format uloží výsledné hodnoty do argumentů. Pro posílání a vyčítání dat tedy byly použity řetězcové funkce, na rozdíl od znakových. Po provedení komunikace je sériová komunikace uzavřena funkcí *fclose()*.

Aby bylo možné funkce zkompileovat, musí být vložen hlavičkový soubor standardní knihovny C <stdio.h>

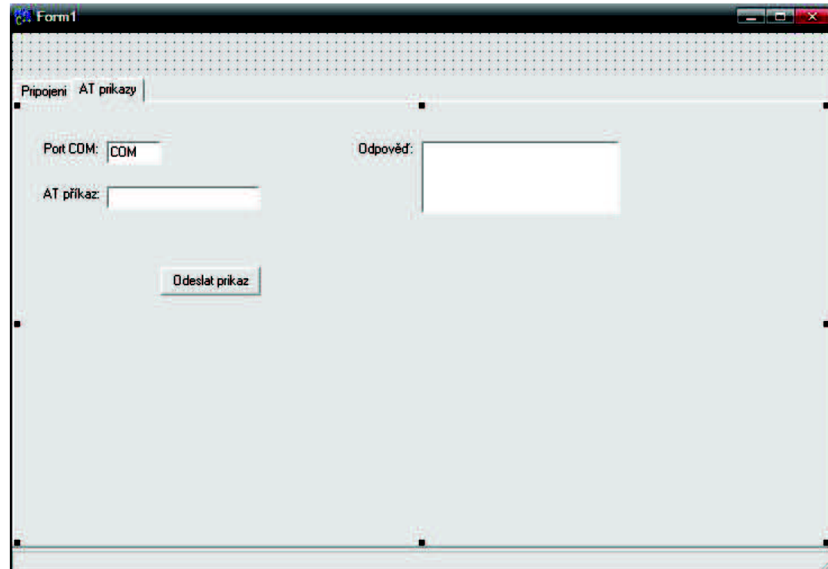
```
#include <stdio.h>

sprintf(comport, "%s", vstupni_data); /* funkce pro poslání řetězce
do sériového kanálu z proměnné vstupni_data, „s“
značí formát řetězec */
sscanf(comport, %s, vystupni_data); /* funkce pro získání řetězce
ze sériového kanálu a uložení proměnné
```



```
vystupni_data, „s“ značí formát řetězec */  
fclose(comport);
```

Při použití vizuálního vývojového prostředí je možné vkládat AT příkazy, číslo portu pomocí prvku Edit a stejně tak je i zobrazovat, viz obr. 4.4. Zdrojový kód programu pak může být napsán pouze v obsluze tlačítka s doplňujícím deklarováním proměnných.



Obr. 4.4: Náhled vizuálního programu posílajícího AT příkazy

Tento postup je dobrá možnost, jak komunikaci vyzkoušet, ale nenabízí příliš komfortu a pro další používání by byla poměrně omezená. Z tohoto důvodu byla pozornost zaměřena na vytvoření databáze AT příkazů, kde by i byl popsán způsob použití a jejich volba by probíhala pomocí polohovacího zařízení. K tomuto by bylo potřebné, aby bylo možno řadit příkazy do fronty pro vykonávání, neboť úkonů potřebuje pro kompletní vykonání několik příkazů.

# 5 PROGRAM PRO MĚŘENÍ KVALITY A DOSTUPNOSTI GSM SÍTĚ

## 5.1 Verze programu

Pro vhodné rozvržení postupu práce na vytvářeném programu bylo zvoleno rozlišení dokončené práce a budoucí práce pomocí verzí. Verze, která je aktuální v době vydání této diplomové práce je 1.1.0.

- Verze 0.1.0 – Naprogramování sériové komunikace přes rozhraní Bluetooth a vytvoření ovládacího rozhraní. Aplikace komponenty ComPort Library.
- Verze 0.2.0 – Vytvoření základního uživatelského rozhraní pro ovládání programu a definování umístění prvků.
- Verze 0.3.0 – Vytvoření databázového systému, kde budou uloženy AT příkazy a výsledky jednotlivých testů.
- Verze 0.4.0 – Vytvoření systému pro tvorbu grafů, kde zdroj dat bude zastávat vybraná databáze změřených dat.
- Verze 0.5.0 – Vytvoření rozhraní pro plánování testů a naprogramování funkčnosti, včetně ukládání souborů.
- Verze 0.6.0 – Aplikace komponenty RAS manageru pro výběr připojení Dial-up a naprogramování sledu AT příkazů pro ověření komunikace (načítání parametrů mobilního zařízení).
- Verze 0.7.0 – Vytvoření rozhraní pro testování parametrů sítě a vytvoření rozhraní pro vkládání testů do plánovače testů.
- Verze 0.8.0 – Vytvoření testu DSS – Doba sestavení spojení a zařazení do rozhraní pro testování.
- Verze 0.8.1 – Vytvoření testu DOS – Doba odeslání SMS a zařazení do rozhraní pro testování.
- Verze 0.8.2 – Vytvoření testu PIN – PING a zařazení do rozhraní pro testování. Aplikace komponenty PING (ICS - Internet Component Suite).
- Verze 0.8.3 – Vytvoření testu DPW – Doba přístupu WWW a zařazení do rozhraní pro testování.
- Verze 0.8.4 – Vytvoření testu PRD – Přenosová rychlost (Dial-up) a zařazení do rozhraní pro testování. Aplikace komponenty HTTPGet.
- Verze 0.8.5 – Vytvoření vlastního testu pro nezávislé ověřování parametrů a nastavení mobilního zařízení pomocí AT příkazů.
- Verze 0.9.0 – Vytvoření informační záložky o programu a slazení postupu při ovládání programu.
- Verze 1.0.0 – Funkční všechny ovládací prvky a také principiální funkčnost vytvořených testů.

- Verze 1.1.0 – Grafické uspořádání komponent a vizuální oddělení částí pro přehlednost programu, doplnění ikon pro lepší přehlednost při ovládání programu a vytvoření nápovědy.
- Verze 1.2.0 – Odstranění chyb, které se vyskytují při běhu programu a ošetření událostí, které by mohly vést k nestandardním událostem. Doplnění funkce tlačítek rychlé volby a naprogramování jejich obsluhy, doplnění horního uživatelského menu a naprogramování obsluhy jednotlivých prvků, doplnění dolního panelu pro zobrazování statusu.
- Verze 1.3.0 – Doplnění programu o možnost testovat více operátorů zároveň pomocí více připojených mobilních zařízení.
- Verze 1.4.0 – Přidání funkce testování pro automatickou ICQ komunikaci mezi mobilními zařízeními.

## 5.2 Jednotlivé části programu

Snahou při vytváření programu bylo, aby byl přehledný pro uživatele a přesto skýtal velké možnosti nastavení parametrů. Proto je ovládání rozděleno do několika sekcí (záložek) pomocí komponenty PageControl, kde každá z nich je určena pro určitou práci s programem. I když jsou části od sebe vizuálně odděleny, data vytvořená v jednotlivých procedurách jsou přístupná globálně a tak lze např. hodnoty z jednoho testu v jedné části prohlížet a v druhé zpracovávat ve formě grafu nebo exportovat.

I když záměr při vytváření programu bylo maximální zjednodušení a přehlednost pro uživatele, mohou některé sekce zdržovat při vytváření sledu testování, protože je uživatel musí mezi nimi stále přepínat.

Kromě střední části, kde se nachází komponenta PageControl, je na dolní hraně okna programu vložena komponenta StatusBar. Do programu jsou přidány i komponenty ToolBar a MainMenu. Ačkoliv v nynější verzi nejsou komponenty plně funkční, nikterak nebrání chodu programu či nijak vizuálně neruší. Jejich doprogramování bylo plánováno ve verzi 1.2.0.

### 5.2.1 Záložka Připojení zařízení

Aby bylo možné testovat jednotlivé vlastnosti sítě a získávat data z testů, je nutné, aby program komunikoval s mobilním zařízením. Mobilní zařízení je možné připojit přes rozhraní Bluetooth nebo USB. Využívá se tedy sériové komunikace a zařízení se vybírá pomocí portu COM, ke kterému je přiřazeno. Pro komunikaci je nutné nastavit i další parametry, kde jeden z nejdůležitějších je přenosová rychlost. Ta se u mobilních zařízení různí a při chybném nastavení budou data buď přenášena pomalu, což zkresluje výsledky některých testů, nebo zařízení nebude dostatečně rychlé k vytvoření odpovědi na dotaz a data nebudou odesílána korektně. Tento problém však nastává pouze u starších modelů mobilních zařízení.

Kromě připojení mobilního zařízení k sériovému portu COM je pro některé testy nutno určit i připojení Dial-up v RAS manageru. Jedná se hlavně o případ, kdy je počítač připojen k více zdrojům internetu (síťový kabel, Wi-Fi). Pokud by byly data přenášeny jinou cestou, např. při testování přenosové rychlosti, než je

připojení k internetu přes mobilní zařízení, mohly by být výsledky testu značně zkráceny.

Pro to, aby bylo možné mobilní zařízení využívat programem, je nutná předchozí instalace v operačním systému. Pokud nebude mobilní zařízení korektně nainstalováno, nebude ho možné v programu určit, protože se neobjeví v seznamu použitelných portů COM a ani v seznamu RAS manageru. Tyto seznamy jsou načítány z registrů systému Windows samotnými komponentami v programu.

Záložka Připojení zařízení je rozdělena na dvě části pomocí komponent GroupBox\_Param a GroupBox\_Zariz. V pravé části (v komponentě GroupBox\_Zariz) se nachází test sériové komunikace. Test se skládá z odesílání AT příkazů a zpracování odpovědi mobilního zařízení a slouží k ověření správného připojení mobilního zařízení k portu COM. Výsledkem tohoto testu jsou základní informace o mobilním zařízení, jako je jméno výrobce, model mobilního zařízení, výrobní série mobilního zařízení, sériové číslo mobilního zařízení, identita, lokální čas, stav baterie, úroveň signálu, ID buňky, ke které je mobilní zařízení připojeno a jména operátorů, které mobilní zařízení má „v dohledu“, zobrazovaných v jednotlivých polích Edit, které jsou povoleny pouze pro čtení. Příkazy jsou vysílány postupně za sebou, takže je nejprve odeslán AT příkaz přes sériový port. Poté je do programu vloženo čekání, buď pomocí vlastního podprogramu wait() nebo pomocí Slep() na zhruba 200ms, protože mobilní zařízení potřebuje určitý čas na zpracování a odeslání odpovědi, který se různí. Poté je čtena příchozí odpověď, upravena a zobrazena v poli, viz následující ukázka zdrojového kódu. Před dalším odesláním příkazu je do programu vložena další čekací smyčka. Aby nebyly mobilním zařízením vráceny odeslané příkazy, je použit příkaz ATE0 pro vypnutí echa. Pro zapnutí a také opakování testu připojení slouží tlačítko Refresh (BitBtn\_Refr), to je vhodné, pokud některá informace nebyla načtena z mobilního zařízení korektně.

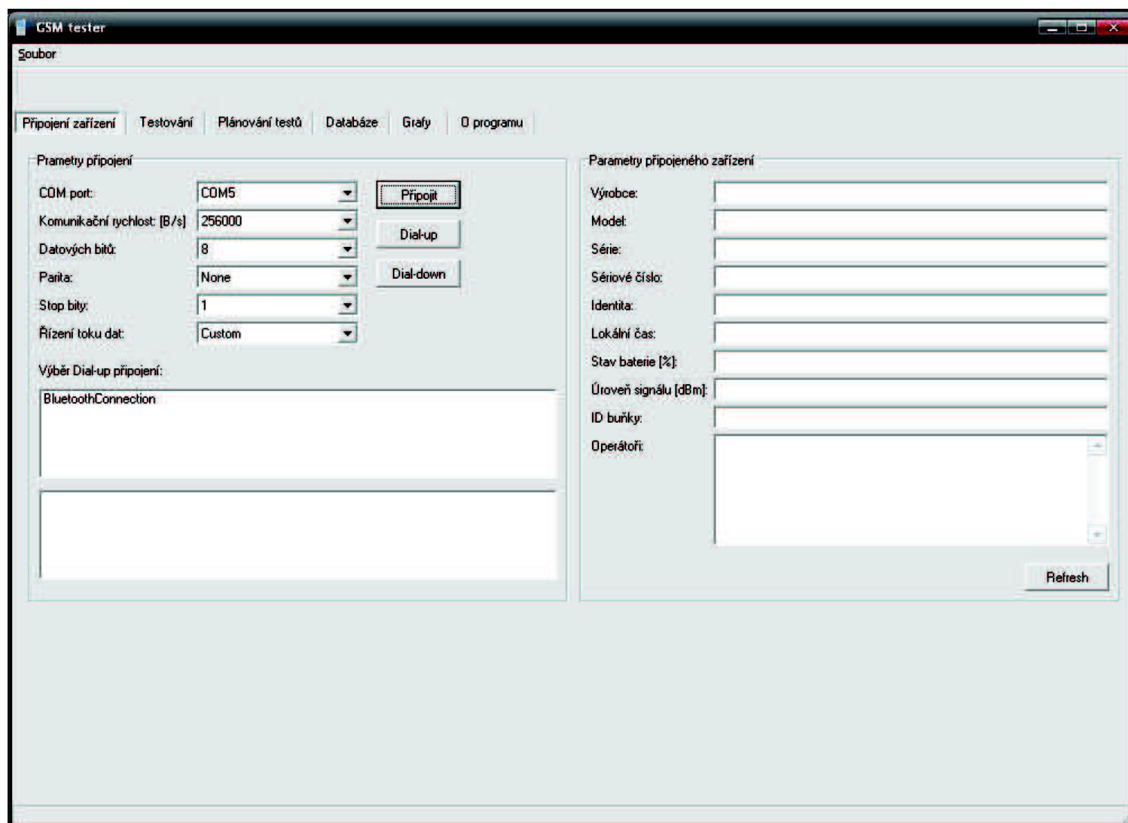
```
AnsiString Str = 0;
AnsiString Odp = 0;
char *OdpCh;
char *pom;

ComPort->WriteStr("ATE0\r");
wait(0.5);
ComPort->ReadStr(Odp,100);
wait(0.5);

Str = "AT+CGMI\r";
ComPort->WriteStr(Str);
wait(0.5);
ComPort->ReadStr(Odp,100);
wait(0.5);
OdpCh = Odp.c_str();
pom = strtok(OdpCh, "\\r\\n");
Edit_CGMI->Text = pom;
```

K připojení k mobilnímu zařízení na portu COM slouží tlačítko Připojit (BitBtn\_COM), které se po připojení automaticky změní na Odpojit. Před připojením je ale vhodné nastavit parametry pro komunikaci, mezi kterými je výběr portu, komunikační rychlost, počet datových bitů, parita, počet stop bitů a řízení toku dat. K připojení pomocí RAS manageru slouží tlačítko Dial-up (BitBtn\_Dial) a k odpojení je určeno tlačítko Dial-down (BitBtn\_Hang). Seznam možných připojení je zobrazen v prvním informačním okně (Výběr Dial-up připojení,

ListBox\_Sez). Po výběru žádaného připojení a kliknutím na tlačítko Dial-up je v druhém informačním okně zobrazen průběh a status připojení (Memo\_Ins), viz obr. 5.1. Vytvoření popisovaných funkcí bylo vytvořeno pomocí komponent třetích stran RasManager a ComPort. Komponenty jsou primárně určeny pro Delphi, ale soubor \*.pas lze využít i v C++ Builderu.



Obr. 5.1: Náhled na záložku Připojení zařízení

## 5.2.2 Záložka Testování

Hlavní účel, ke kterému program byl a je vytvářen je testování parametrů sítě GSM. Nástroje pro testování jsou tedy ukryty v záložce Testování, viz obr. 5.2, která navazuje hned za záložkou Připojení zařízení. Po připojení mobilního zařízení můžeme tedy zahájit vlastní testování.

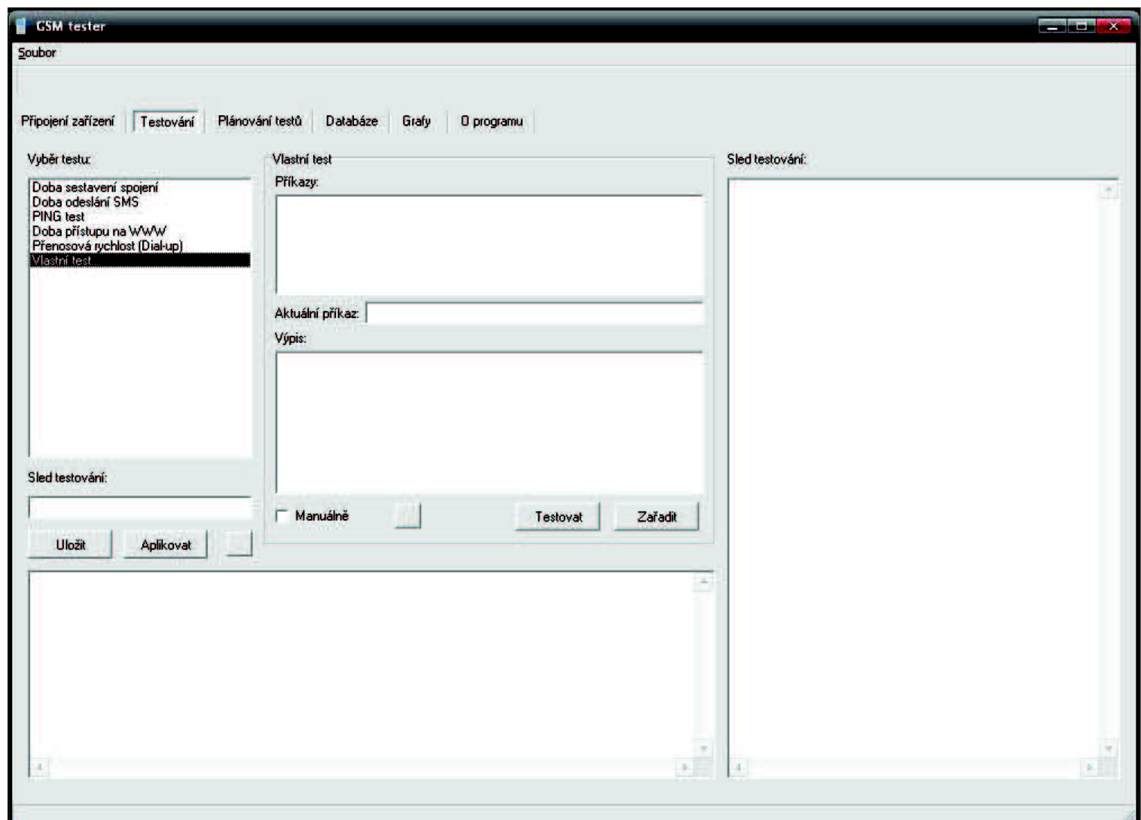
V List\_Box\_Tst pod popisem Výběr testu, je uloženo několik základních testů, kterými program disponuje. Jedná se o dobu sestavení spojení, dobu odesílání SMS zprávy, test PING, dobu přístupu k internetové stránce, přenosovou rychlost stahovaného souboru z internetu a vlastní test, který si uživatel sám definuje pomocí AT příkazů z databáze. Po vybrání testu se vpravo od list boxu objeví položky pro nastavení daného testu. Tyto položky jsou popisovány v následujících kapitolách k popisu určeným.

Ve vybraném testu má uživatel možnost testovat okamžitě, po kliknutí na tlačítko Testovat, nebo zařadit test do fronty tlačítkem Zařadit. V případě volby testovat se provede test dle zadaných kritérií a výsledek se uloží do textového souboru. V případě volby zařadit se provede zápis do sledu testování.

Sled testování je seznam (ListBox\_Test), do kterého se postupně řadí testy i se zvolenými parametry, aby mohly být později provedeny dle plánu testování. Formát zápisu začíná vždy zkratkou testu, poté jeho jménem a poté následují parametry pro testování, např. DSS – Doba sestavení spojení. Tel. Číslo: 123456789. Opakování 10x.

Aby uživatel nemusel pokaždé vytvářet nový sled testování, může si jej načíst z textového souboru. Po kliknutí na pomocné tlačítko SpeedButton\_Sled se vyvolá dialogové okno OpenFileDialog pro otevření textového souboru. Po potvrzení je textový soubor automaticky otevřen a načten do komponenty Memo\_Nacten umístěné vlevo dole a v Edit\_Sled se objeví cesta k načtenému souboru. Pokud načtený seznam připadá uživateli vhodný, může jej aplikovat tlačítkem Aplikovat (BitBtn\_Aply). Poté se načtený seznam přesune do sledu testování a smaže se z přehledového okna. Seznam je možné v náhledovém okně i upravovat.

Vytvořený sled testování je možné taky uložit do textového souboru po kliknutí na tlačítko Uložit (BitBtn\_Uloz). Po kliknutí se vyvolá dialogové okno pro uložení SaveDialog a pomocí něho se určí umístění a název souboru. Přípona \*.txt je přidána za jméno automaticky.



Obr. 5.2: Náhled na záložku Testování

### 5.2.3 Záložka Plánování testů

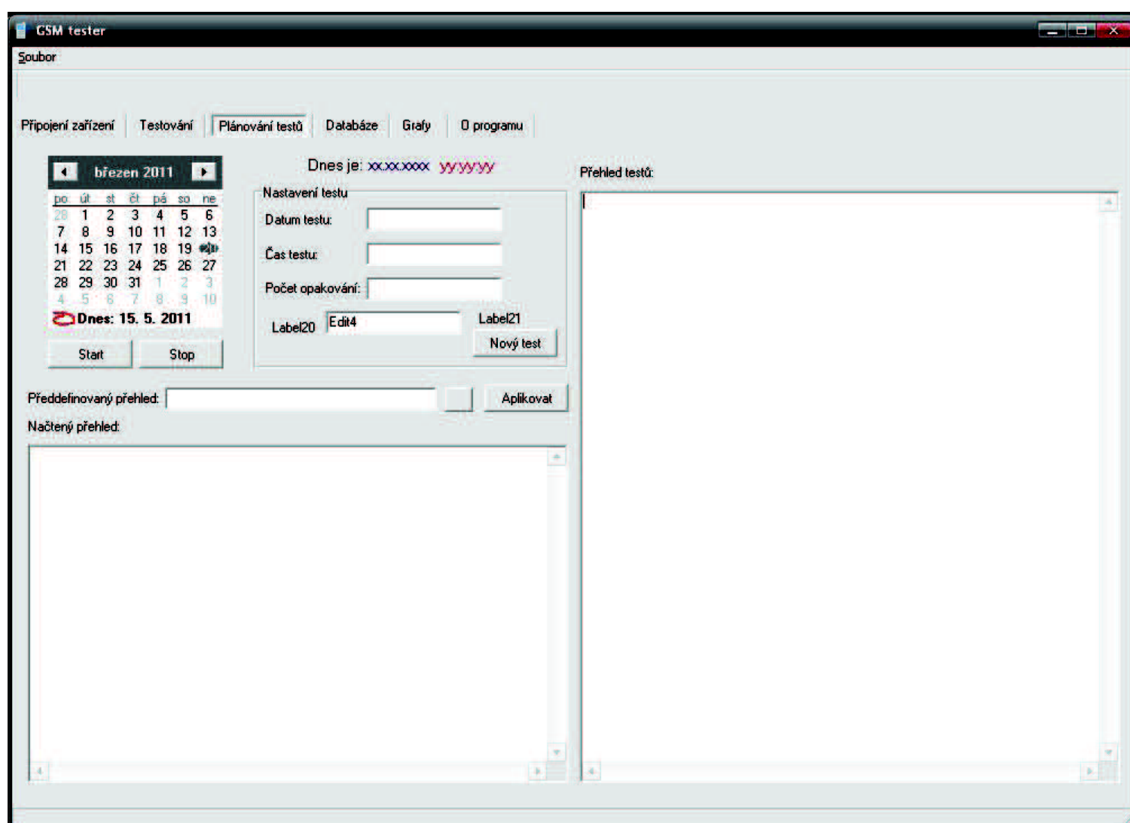
Parametry sítě GSM jsou poměrně proměnlivé. Vliv na proměnlivost parametrů má na svědomí proměnlivé okolní rušení nebo i povětrnostní podmínky. Ty se mění i s dobou a z tohoto důvodu byl začleněn do programu plánovač testů. Obsluha tak kromě využívání jednorázových testů bude moci využít naplánované testy a nebude muset stále kontrolovat chod testů.

Stejně jako předchozí záložka Testování nabízí záložka Plánování testů načtení a uložení přehledu testů. Výhodnost je jednoznačná, neboť uživatel nebude muset opakovat zadávání testů, které mohou být mnohdy rozsáhlé. Test je uložen ve formě textového souboru a je možné načíst pomocným tlačítkem (SpeedButton\_StarTes). Po načtení se automaticky objeví v přehledovém oknu (Memo\_Nact), kde je ho možné ještě dodatečně upravit a poté aplikovat kliknutím na tlačítko Aplikovat (BitBtn\_Apl).

Pokud uživatel bude vytvářet nový test, musí zadat datum (k tomu lze využít i přidružený kalendář) a čas začátku testu, případně počet opakování. Po kliknutí na tlačítko Nový test (BitBtn\_Novy) se objeví v Přehledu testů položka, která bude označovat začátek nového velkého testu. Pod ní se pak budou řadit jednotlivé testy, které se vyberou na záložce Testovat volbou tlačítka Zařadit. Po dokončení volby jednotlivých malých testů lze vložit další velký test kliknutím na tlačítko Nový test.

Po dokončení voleb pro testování stačí kliknout na tlačítko Start (BitBtn\_Start). Po jeho aktivaci dojde k porovnání data a času prvního testu a lokálního času. Pokud nastane shoda, nastaví se počet opakování a započne testování. Test se rozliší svou zkratkou na počátku a podle jeho zkratky dojde k vyčtení parametrů, které jsou dále přiřazeny dílčímu testu. Ten se spustí a při jeho funkci jsou jeho výsledky ukládány do databáze, která je při jeho zahájení vytvořena. Po dokončení program přistoupí k dalšímu testu.

Ke zrušení testování dojde po kliknutí na tlačítko Stop (BitBtn\_Stop). Pokud je malý test přerušen, nebude již dále navazovat a zůstává tak nedoměřen. Pokud test probíhá, je tato událost indikována nápisem.



Obr. 5.3: Náhled na záložku Plánování testů

## 5.2.4 Záložka Databáze

Pro přehlednost výsledků jednotlivých testů jsou data ukládány do databáze pod názvem ve formátu názevtesturokměsícdenhodinaminutasekunda. Pro každý provedený test je tedy vytvořena nová databázová tabulka s daty, které je možno prohlížet, exportovat do MS Word nebo MS Excel nebo zpracovávat ve formě grafu na následující záložce.

Kromě tabulek s výsledky testů je zde tabulka AT příkazů, která slouží pro tvorbu vlastního testu. Některé AT příkazy mohou mít několik parametrů, a proto je v ohraničené oblasti Parametry AT příkazů (GroupBox\_ParAT) vytvořeno několik komponent Edit, do kterých se parametry zapisují. Počet komponent Edit je zobrazen podle načteného AT příkazu. Pokud chce uživatel příkaz vložit do vlastního testu, může kliknout na tlačítko Zadat (BitBtn\_Zad). Jako obsluha tlačítka se provede otestování vyplněných položek, vytvoření celkového příkazu a zápis do vlastního testu. Po úspěšně provedené akci se vedle tlačítka objeví zelený nápis Zapsáno.

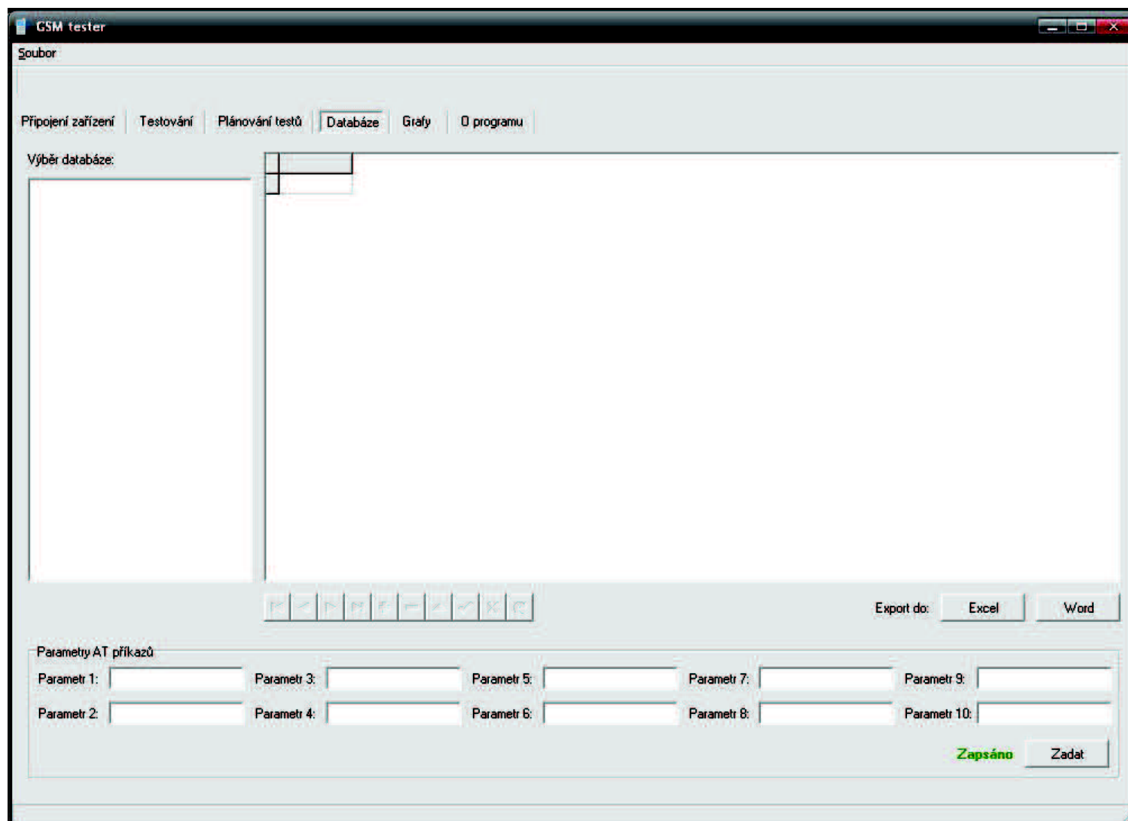
Pro potřeby dalšího zpracování jsou umístěny pod zobrazovací komponentou BDGrid tlačítka pro export dat do MS Word nebo MS Excel (BitBtn\_Exc, BitBtn\_Wor). Po kliknutí na jedno z tlačítek se vytvoří nový dokument, do kterého se vloží data z vybrané databáze. Pro vytvoření dokumentu je nutné, aby byly na PC nainstalovány programy sady MS Office Excel a Word. Vytvoření nového dokumentu by nemělo být závislé na aktuální verzi kancelářských nástrojů, nelze však zaručit plnou kompatibilitu v rozdílných verzích, protože vydání programovacího nástroje C++ Builder je z roku 2002.

Jak je vidno z obr. 5.4, ovládací rozhraní se neskládá jen z polí Edit a tlačítek pro export. Hlavní částí záložky databáze je zobrazovací komponenta pro databázi DBGrid, ve které se zobrazí vybraná tabulka databáze po vybrání v levém seznamu databází ListBox\_Vyb. Do výběru databáze se vždy přidávají nové záznamy postupně pod sebe, jak jsou prováděny naplánované testy. K ovládní databáze, tedy přidávání, odebírání a posouvání záznamů slouží komponenta DBNavigator umístěná pod komponentou DBGrid.

Samotné přidání komponenty, pro zobrazení databáze, nestačí. Aby program plně pracoval s databází, je tedy nutno nejprve vytvořit vlastní databázi, resp. nový alias, např. pomocí doplňku k C++ Builderu BDE Administrator. V něm se nastaví databázový ovladač, který je ponechán na položku Standard. Po vytvoření nového aliasu a zadání cesty k databázové složce je možno přistoupit k vytvoření nové databáze. Nová databáze lze snadno vytvořit v dalším doplňku C++ Builder Databáze Desktop (v něm je případně možné i vytvořit nový alias bez nutnosti používat BDE Administrator). Při vytváření nové databáze je nutné zvolit, o jaký typ databáze se bude jednat. Nejběžnější je databáze typu Paradox 7. Poté už je jen nutno vytvořit uspořádání databázové tabulky (sloupce a popisy) a přiřadit databázi k dříve vytvořenému aliasu.

Do programu se databáze přiřadí pomocí komponenty Table. V ní se zvolí vytvoření alias a z něho se vybere databázová tabulka, se kterou se bude pracovat. Aby bylo možné data zobrazovat v komponentě DBGrid a pracovat s nimi pomocí komponenty DBNavigator, je nutné přidat komponentu DataSource, které zprostředkuje přístup ke zvolené databázi.





Obr. 5.4: Náhled na záložku Databáze

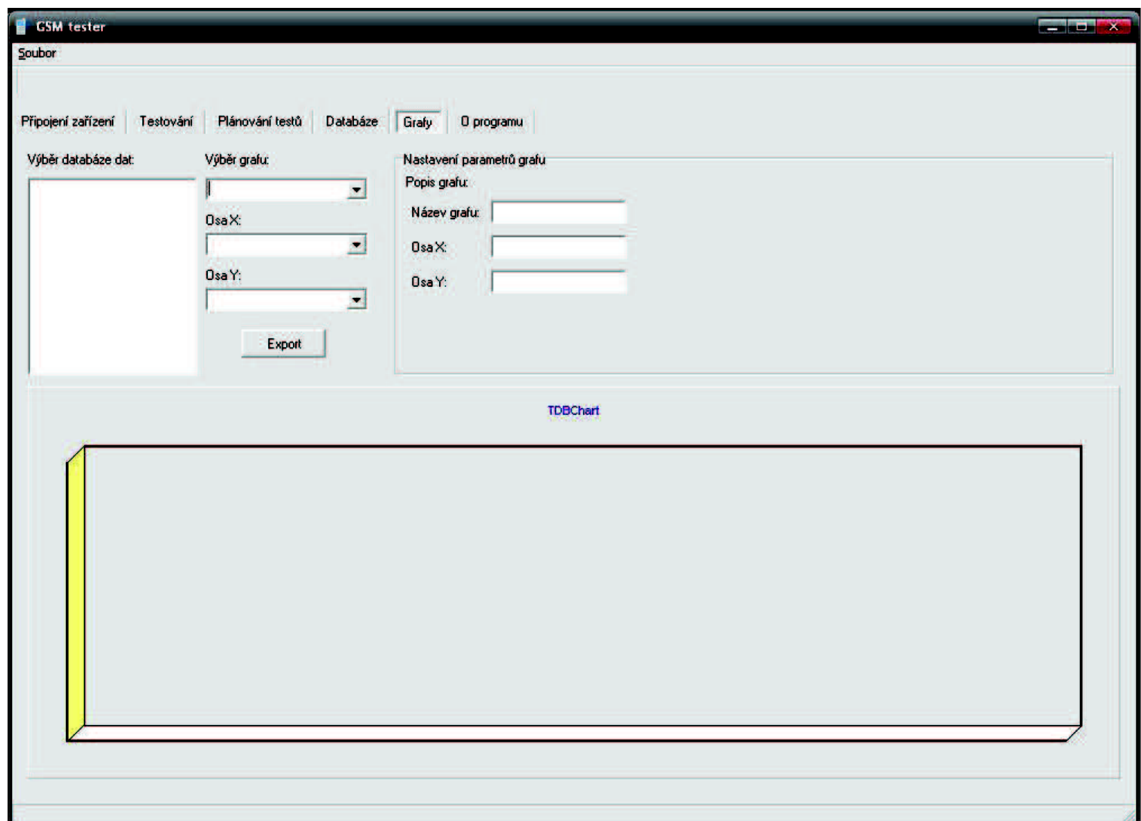
### 5.2.5 Záložka Grafy

Pro vyhodnocení testu ve formě grafu slouží záložka Grafy. Vytvářet grafy lze jen z dat uložených v databázi, takže se jedná o výsledky jednotlivých naplánovaných testů.

Z levého okna, kde je umístěn seznam proběhlých testů (ListBox\_VybD) se pomocí myši vybere jeden z testů. Druhou volbou je volba grafu, který se má zobrazit. V závislosti na testu se poté vyberou sloupce dat, které mají být k jednotlivým osám grafu přiřazeny. Podle volby grafu nemusí být obě osy zastoupeny (např. volba výšechového grafu). S volbou grafu jsou zobrazeny i další volitelné parametry, jako je název grafu, pojmenování os apod.

Ústřední komponentou této záložky je komponenta grafu DBChart. Její zdroje a vlastnosti jsou přímo spjaty s používanou databází v záložce Databáze. Stejněho typu jsou i komponenty pro výběr grafu a dat k jednotlivým osám.

Poslední položkou v této záložce je tlačítko Export (BitBtn\_ExpG), které vytvoří obrázek vyobrazeného grafu.

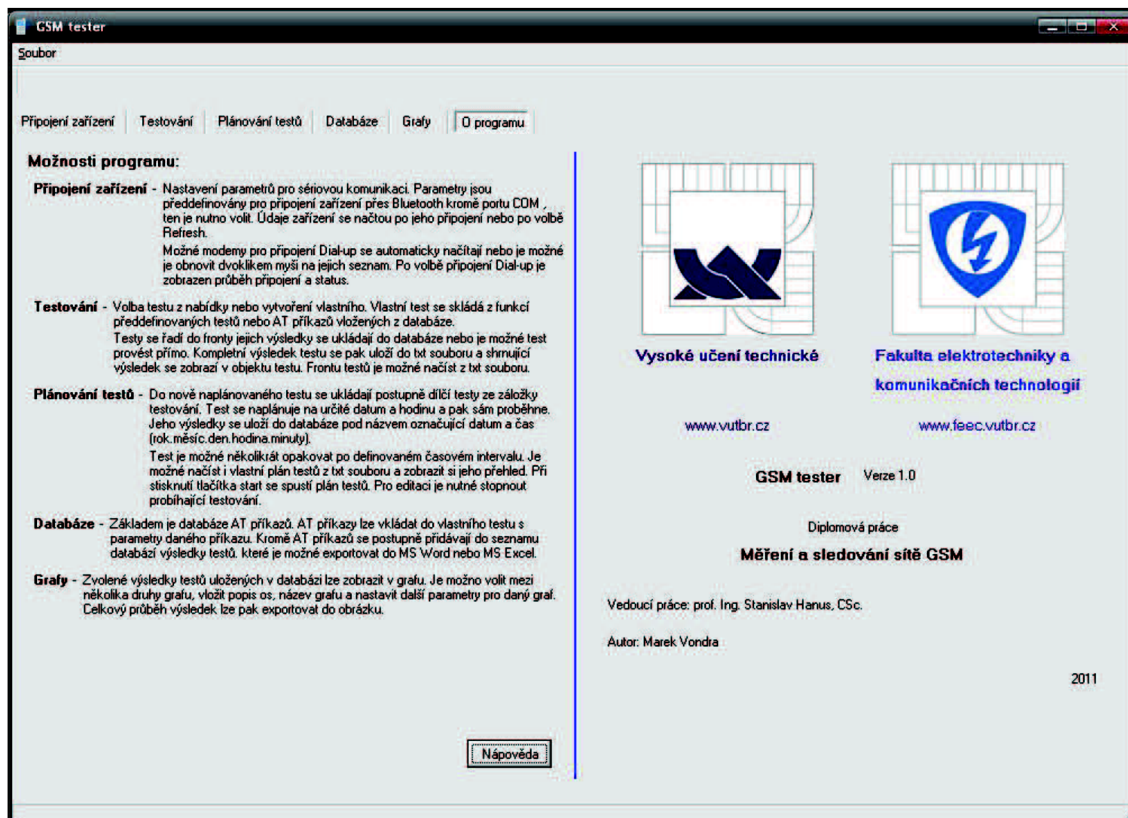


Obr. 5.5: Náhled na záložku Grafy

### 5.2.6 Záložka O programu

Záložka je situována do dvou částí oddělených komponentou Splitter. V levé části je několik komponent Label s povolením pro zalamování řádků, které tvoří stručný popis předchozích záložek a nápovědu k nim. Pro vyvolání kompletní obsáhlé nápovědy slouží tlačítko Nápověda (komponenta BitBtn\_Nap). Nápověda je ve stejném adresáři s programem pod názvem Napověda.hlp a je do programu přidána přes pomocnou nabídku v C++ Builderu Project – Options – Application – Help file.

V pravé části záložky jsou umístěny komponenty Image\_VUT a Image\_FEKT pro zobrazení log Vysokého učení technického v Brně a Fakulty elektrotechniky a komunikačních technologií. Pod nimi je situováno několik komponent Label s neaktivními internetovými odkazy, názvem a verzí programu, tématem diplomové práce, jménem vedoucího práce a jménem autora, viz obr. 5.6.



Obr. 5.6: Náhled na záložku O programu

## 5.2.7 Nápověda

Program je od počátku tvořen, aby ovládání bylo intuitivní, ale přesto se mohou objevit situace, kdy uživatel nebude natolik znalý, aby program použil bez pomoci nápovědy. Pro tento účel byla vytvořena nápověda a přidána k programu (viz kapitola Záložka O programu).

Samotné vytvoření nápovědy je možné buď pomocí C++ Builderu nebo pomocí některého ze specializovaných programů. Výsledkem je vždy soubor s příponou \*.hlp.

Pokud je nápověda vytvářena pomocí C++ Builderu, je nutno nejprve vytvořit textový soubor s příponou \*.rtf v editoru, který umožňuje vytvářet poznámky pod čarou, dvojitě podtržený text a podporuje skryté znaky. Po vytvoření textového souboru je nutné vytvořit soubor projektu, což je obyčejný textový soubor s přidáním příponou \*.txt. V těle textu je nutná deklarace, viz příklad zdrojového kódu.

```
[OPTIONS]
TITLE= // Text v záhlaví nápovědy
OEFONT= // Písmo
CONTENTS= // Kontextový řetězec hlavního tématu
COMPRESS= // Stupeň komprese souboru. Vhodné nastavit High.
ERRORLOG= // Soubor pro ukládání chybových hlášek vzniklých při překladau
```

```
[FONTS]
Arial=Arial CE // Použité fonty.
```

```
[FILES] // Uvedení souborů *.RTF
Soubor1.RTF
```

```
[MAP] // Mapování kontextových řetězců na čísla použitelná v C++
Builderu
Retězec1 1...
Retězec2 2...
```

Poté už stačí náповědu zkompileovat v některém z kompilátorů. U C++ Builderu je pomocný kompilátor Microsoft Help Workshop.

## 5.3 Testovací prostředky

Testování sítě GSM je prováděno pomocí předprogramovaných testů nebo pomocí vlastního testu, který umožňuje využít pouze AT příkazů. Nalézají se v záložce Testování a jedná se o test doby sestavení spojení, doby odesílání SMS, testu PING, doby přístupu k internetové stránce, přenosové rychlosti a vlastní test.

Každý test je možné provádět buď ihned nebo zařadit do fronty pro testování. Pokud se provede ihned, bude jeho výstupem textový soubor, který je pojmenován zkratkou testu a pak složením řetězce s číslem roku, měsíce, dne, hodiny, minuty a vteřiny, viz následující zdrojový kód. K vytvoření názvu souboru se používá lokální čas počítače, takže při špatném nastavení času budou vznikat i zkreslené názvy testování. Po vytvoření souboru je soubor uložen do aktuálního adresáře programu.

```
time_t rawtime; // casova struktura
struct tm * timeinfo;
char buffer_time [21]; // deklarace pole pro uchovani znaku
time ( &rawtime );
timeinfo = localtime ( &rawtime ); // nacistani lokálního casu
strftime (buffer_time,21,"DSS%y%m%d-%H%M%S.txt",timeinfo);
handle = FileCreate(buffer_time); // vytvoreni souboru
FileSeek(handle, 0, 2); // nastaveni kurzoru na pocatek souboru
buffer = "TEST: Doba sestavení spojení - ";
strftime (pobuf,42,"Čas měření: %d.%m.%Y
%H:%M:%S\r\n\r\n",timeinfo);
strcat(buffer, pobuf);
delka = strlen(buffer); // delka retezce k zapisu
FileWrite(handle, buffer, delka); // zapis do souboru

FileClose(handle); // uzavreni souboru
```

V ukázce zdrojového kódu je obsaženo i vytvoření hlavičky textového souboru, v které je obsažen název testu a datum a čas testování. Poté je řetězec zformován a uložen do souboru. Za hlavičkou již následují údaje o měření, které jsou u každého měření jinak formátovány. Po dokončení měření se soubor vždy uzavře a program s ním již přestane pracovat.

### 5.3.1 Test DSS – Doba sestavení spojení

Základní parametr, který běžného uživatele mobilního zařízení zajímá a také jeho vnímání ovlivňuje je doba sestavení spojení. Jedná se o hluché místo při sestavování

hovoru, než započne vyzvánění.

Uživatelská část programu pro měření tohoto parametru se skládá z komponenty Edit pro zadání telefonního čísla (Edit\_Tel), komponenty Edit pro zadání počtu opakování (Edit\_Opak), zobrazovací tabulky pro přehled stávajícího měření StringGrid\_Ses a ovládacích tlačítek Testovat (BitBtn\_Tst) a Zaředit(BitBtn\_Zar), viz obr. 5.7. Po kliknutí na tlačítko Zařadit se vytvoří zápis do sledu testování ve formátu: DSS – Doba sestavení spojení. Tel. Číslo: 123456789. Opakování 10x. Po kliknutí na tlačítko Testovat započne procedura, která nejprve otestuje, zda je mobilní zařízení připojeno k portu COM. Pokud ano, vytvoří se textový soubor s jednotnou hlavičkou a dále se do něj запиše telefonní číslo, na které se bude volat a počet opakování. Poté již následuje netypická procedura dána tomuto měření. Pokud test probíhá, je tato událost indikována nápisem.

Měření doby sestavení spojení je poměrně problematické, protože doba sestavení není přímo dána AT příkazy (není žádný AT příkaz pro zjištění dané doby) a je nutno ji detekovat. Pokud je tedy odeslán AT příkaz pro sestavení hovoru „ATD123456789,“, vytvoří mobilní zařízení spojení, které trvá jen určitý časový interval, který je uložen v registru S8, protože je příkaz zakončen čárkou. Aby mobilní zařízení vytvořilo spojení, je nutné nahradit čárku středníkem. Pokud je spojení možné sestavit, pošle mobilní zařízení odpověď „OK“, pokud ne, odešle odpověď „NO CARRIER“. Poté už mobilní telefon přímo nijak nedetekuje stav, zda spojení bylo sestaveno či nebylo. K tomu je možné využít AT příkaz AT+CIND. Tento AT příkaz v odpovědi zveřejní aktuální stav mobilního zařízení. Mezi těmito stavy je i položka callsetup, který nabývá hodnot od 0 do 3, kde 0 je hovor neaktivní, 1 přichází hovor, 2 odchozí hovor nesestaven, 3 odchozí hovor sestaven. Tento příkaz bývá podporován u všech mobilních zařízení, ale není tomu tak i se stavem callsetup, který u některých mobilních zařízení nemusí být zastoupen. Pokud zastoupen je, jeho umístění se různí, takže je nutno nejprve vyhledat jeho pořadí v seznamu stavů. Po získání jeho pozice je pak možné lehce detekovat výstupní stav.

Nalezení pozice stavu callsetup je provedena tak, že se nepotřebné znaky v řetězci odpovědi (+CIND: ("battchg",(0-5)),("signal",(0-5)),("batterywarning",(0-1)),("chargerconnected",(0-1)),("service",(0-1)),("sounder",(0-1)),("message",(0-1)),("call",(0-1)),("roam",(0-1)),("callsetup",(0-3))) nahradí dvojtečkou, přičemž řetězec odpovědi je odpověď mobilního zařízení na příkaz AT+CIND=?\r. Zbudou tedy jen klíčová slova a stavy, odstraní se tím závorky, uvozovky, čárky a pomlčky. Poté dojde k postupnému rozdělování řetězce, až je nalezen odpovídající hledaný stav. Počet cyklů, než je hledaný stav nalezen, odpovídá pozici, kde se nachází hodnota daného stavu v odpovědi mobilního zařízení (+CIND: 4,5,0,0,1,0,0,0,0,2) na dotaz AT+CIND=?\r, což je v uvedeném případě poslední pozice, která napovídá, že odchozí spojení ještě nebylo sestaveno.

Po nalezení pozice stavu může dojít k samotnému testování, kdy v opakované smyčce for() proběhne postupně několik spojení (podle zadaného počtu), poté se otestuje odpověď, zda mobilní zařízení úspěšně vytvořilo odchozí hovor a poté se v podmínce testuje, kdy se změnil stav z hodnoty 2 na hodnotu 3. Tím se změří doba sestavení spojení. Po jejím změření se hodnota zobrazí v rychlém přehledu, запиše do souboru a vypočítá se její průměr. Po dokončení měření je soubor uzavřen a pomocné proměnné uvolněny z paměti.

Měření doby sestavení spojení je dáno rozdílem dvou časových okamžiků. Jelikož samotný operační systém Windows přistupuje k měření systémového času až po několika ms, bylo nutné provádět přesné měření časových okamžiků tak, že se uloží

hodnoty počtu taktů procesoru, které se počítají od počátku procesu, při začátku a při konci měřeného úseku a poté se vynásobí časovou konstantou procesoru, viz zdrojový kód. Tím je docíleno přesného měření času. Odchylka měření ale bohužel nastává při kontrole změny stavu, protože pro zjištění je nutné vyslat odpovídající AT příkaz, poté počkat na zpracování mobilním zařízením a poté číst odpověď. Při této funkci dochází ke zkreslení měření o hodnotu, která může být v rozmezí intervalu (0;200> ms.

```
clock_t start, stop; // pro mereni casu
start = clock(); // zacatek mereni
stop = clock(); // konec mereni
cas = ((stop - start)/((double) CLK_TCK));
```

Doba sestavení spojení

Tel. číslo:

Počet opakování:

Průběžné výsledky:

Pořadí	První	n - 2	n - 1	Poslední	Průměr
Čas [ms]					

Testovat      Zařadit

Obr. 5.7: Test - Doba sestavení spojení

### 5.3.2 Test DOS – Doba odeslání SMS

Kromě měření doby sestavení spojení je zajímavým parametrem, který uživatele mobilního zařízení může zajímat, doba odeslání SMS zprávy. Jedná se o dobu, která proběhne od odeslání zprávy do přijetí doručky.

Základními prvky pro ovládání testu jsou komponenty Edit pro vyplnění telefonního čísla, na které se bude zpráva SMS odesílat, a počet opakování, kolikrát se bude akce opakovat. Text k odeslání se vypíše do položky Memo\_SMS, která je omezena na 800 znaků, protože jedna zpráva obsahuje maximálně 160 znaků a limit odesílaných zpráv je 5. Více nepůjde jednorázově z programu odeslat. Počet vyplněných znaků a počet zpráv se zobrazuje hned pod oknem s textem.

Aby uživatel nemusel vždy zprávu ručně psát, může si ji načíst ve formě textového souboru kliknutím na pomocné tlačítko (SpeedButton\_Nahr). Po jeho aktivaci se objeví okno s průvodcem otevření souboru a po kliknutí na tlačítko Načíst (BitBtn\_Nahr) se

text z textového souboru načte do okna zprávy. Pokud je větší jak 800 znaků, tak se automaticky omezí.

Po kliknutí na tlačítko Testovat (BitBtn\_TstSMS) se pomocí AT příkazů nastaví telefon do textového módu a potom se mu zašle formát SMS zprávy k odeslání obsahující telefonní číslo a text s prvními 160 znaky. Dalších 160 znaků se vysílá stejným způsobem, ale bez nastavování módu telefonu. To se děje do doby, než je ukončen program počtem opakování. Pokud test probíhá, je tato událost indikována nápisem. Průběžné výsledky jsou zobrazovány v přehledové tabulce a zaznamenávány do textového souboru, který obsahuje standardní hlavičku, telefonní číslo, počet opakování, odesílaný text a jednotlivé výsledky.

Měření doby je dáno deklarací pro rozdíl dvou časů (viz předchozí podkapitola). Začátek měření je umístěn hned za odeslání AT příkazu s odesílající zprávou mobilnímu zařízení a konec je dán rozpoznáním stavu přijetí doručky. To probíhá pomocí čtení událostí mobilního zařízení. Čtení se provádí stále dokola, dokud není zaznamenána odpovídající změna. Mezi jednotlivé čtení parametrů musí být vloženo čekání, aby mohlo mobilní zařízení data zpracovat. Vzniká tak chyba měření, které je dána hlavně dobou čekání. Odchylka může být kolem 200 ms.

Po kliknutí na tlačítko Zařadit (BitBtn\_ZarSMS) se test zařadí do sledu testování tak, že za zkratkou testu je obsaženo telefonní číslo, počet opakování a text zprávy.

Doba odeslání SMS

Tel. číslo:

Počet opakování:

SMS zpráva:

Počet znaků: 0      Počet zpráv: 0

Načíst zprávu:

Průběžné výsledky:

Pořadí	První	n - 2	n - 1	Poslední	Průměr
Čas [ms]					

Obr. 5.8: Test - Doba odeslání SMS

### 5.3.3 Test PIN – PING

Test PING je vytvořen za pomoci komponenty třetí strany ICS - Internet Component Suite, která obsahuje část pro testování PING.

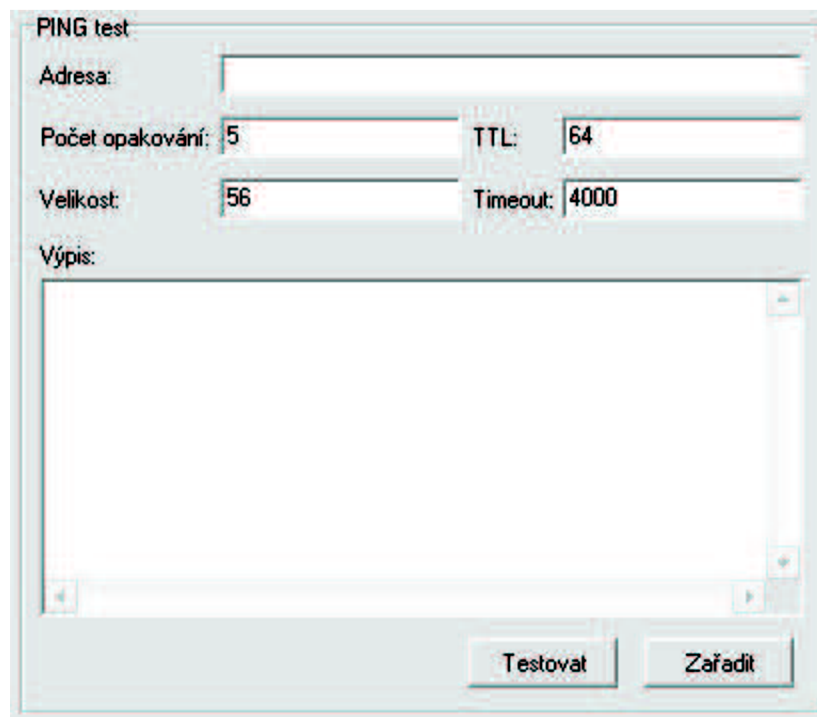
Po kliknutí na tlačítko Zařadit (BitBtn\_PITst) je do sledu testování přidán řetězec začínající zkratkou a názvem testu a dále pak pokračující testovanými parametry, kterými jsou hodnoty TTL, timeout, velikost, počet opakování a cílová adresa. Zápis pak může vypadat např. takto: PING - PING. Opakování: 5x. Adresa: www.feec.vutbr.cz TTL: 64 Velikost: 56 Timeout: 4000.

Při kliknutí na tlačítko Testovat (BitBtn\_PIZar) se provádí test okamžitě. Nejprve jsou nastaveny parametry testování, které se nastavují pomocí komponent Edit, viz obr. 5.9. Poté se volá vlastní podprogram komponenty PING, kdy dojde ke kontrole, zda zadaný cíl existuje. Pokud ano, provede se vlastní testování, viz zdrojový kód. V něm se volá další podprogram komponenty, kdy je testováno, zda lze na zadaný cíl provést PING či nikoliv.

```
for (int i=1; i <= StrToInt(Edit_PIOpak->Text); i++){
    Ping->Address = Ping->DnsResult; //DNS
    Ping->Ping(); // testovani
    if (Ping->Reply.Status != 0) { // hledani erroru
        err ++;
    }
    if (Ping->Reply.RTT > max){ // vyhodnoceni casu mereni
        max = Ping->Reply.RTT;
    }
    else if (Ping->Reply.RTT < min){
        min = Ping->Reply.RTT;
    }
    prum += Ping->Reply.RTT;
}
```

Celý průběh je vypisován do komponenty Memo\_PI a je předdefinován tak, aby se co nejvíce podobal typickému zobrazení výpisu v příkazovém řádku operačního systému MS Windows. Aby bylo možné po dokončení operací vyhodnotit test, tedy kolik odeslaných paketů bylo ztraceno a kolik přijato, je zdrojový kód doplněn o pomocné proměnné a výpočetní algoritmy, které výpočet definují, protože komponenta samotná umí provádět pouze odesílání zpráv a přijímání odpovědí testu PING. Po provedení všech akcí je nakonec vytvořen textový soubor se standardní hlavičkou a pojmenováním a do něj uložen výpis testu.





Obr. 5.9: Test - PING

### 5.3.4 Test DPW – Doba přístupu WWW

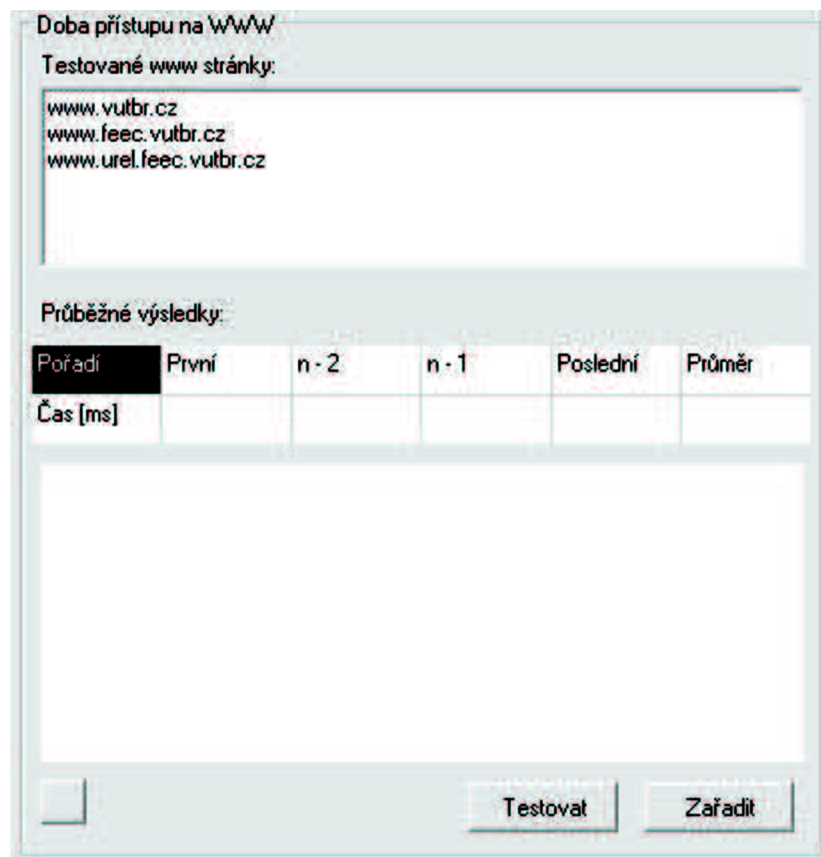
Pro běžného uživatele je jistě podstatné, za jak dlouho se mu zobrazí internetové stránky, které si chce ze svého mobilního zařízení prohlédnout. Pro zhodnocení tohoto hlediska byl vytvořen test pro měření přístupové doby k internetovým stránkám.

Ústřední částí je komponenta internetového zobrazovače CppWebBrowser. Ta slouží k zobrazení zadané internetové stránky. Pro její zobrazení využívá jádra prohlížeče Microsoft Internet Explorer. Vzhledem k tomu, že verze Internet Exploreru jsou novější, než verze vývojového nástroje, dochází občas chybám a výpadkům při komunikaci jádra prohlížeče a vloženou komponentou.

Uživatelskými prvky testu jsou seznam internetových stránek, které se postupně budou načítat zapsaných v Memo\_WSez, zobrazovací tabulka postupných výsledků StringGrid\_WWW, internetový zobrazovač CppWebBrowser a ovládací tlačítka Testovat (BitBtn\_WTst), Zařadit (BitBtn\_WZar) a pomocné tlačítko (SpeedButton\_Pom). Pomocné tlačítko slouží ke změně velikosti testového okna, aby bylo možné testovanou internetovou stránku prohlížet ve větší velikosti, viz obr. 5.10.

Akce, vyvolaná na kliknutí tlačítka Zařadit, přidá zápis do sledu testování ve tvaru DPW - Doba přístupu na WWW. Testované stránky: [www.feec.vutbr.cz](http://www.feec.vutbr.cz).

Po kliknutí na tlačítko Testovat se provede rozklad textového řetězce z komponenty Memo\_WSez na jednotlivé internetové adresy, které jsou v několika smyčkách, odpovídacích počtu adres, použity jako cílové adresy, které se postupně načítají. Načtené adresy se zobrazují v internetovém zobrazovači a změřené výsledky se průměrují a poté zobrazují v informační tabulce. Stejně jako v předchozích testech je vytvořen textový soubor se standardní hlavičkou a jménem, do kterého se zapisují výsledky pro jednotlivé testované internetové stránky. Pokud test probíhá, je tato událost indikována nápisem.



Obr. 5.10: Test - Doba přístupu WWW

### 5.3.5 Test PRD – Přenosová rychlost (Dial-up)

Mezi situace, které patří do každodenního dění je stahování souborů z internetu. Pro změření rychlosti stahování souboru pomocí připojení Dial-up je připraven test PRD. Test je založen na vložené komponentě třetí strany s názvem HTTPGet, která umožňuje buď stažení binárních nebo textových dat (vhodné pro html stránky).

Testu stačí zadat adresu s cílovým umístěním souboru v síti internet a kliknout na tlačítko Testovat (BitBtn\_DTest). Poté se objeví dialog pro určení místa uložení souboru a jeho pojmenování. Po určení umístění započne hledání souboru na internetu, pokud je nalezen, načte se jeho hlavička, aby se určila velikost souboru a započne stahování. Průběh stahování je zobrazován pomocí ProgressBar, kde jeho maximum je velikost stahovaného souboru a posun je dán změnou programu při stahování. Pomocí této změny a času lze vypočítat přenosovou rychlost a dobu do konce stahování souboru.

Výpočet přenosové rychlosti je vytvořen pomocí globální deklarace pro měření času a lokálního zápisu při jednotlivých změnách při stahování. Pomocí měření času je odpočítáván čas rovnající se 1 sekundě. Po uplynutí času se určí počet přenesených bitů, což udává přenosovou rychlost bit/s, viz následující zdrojový kód.

```
if (start == 0){
    stav1 = Readed; // velikost souboru pri zacatku mereneho
    intervalu
```

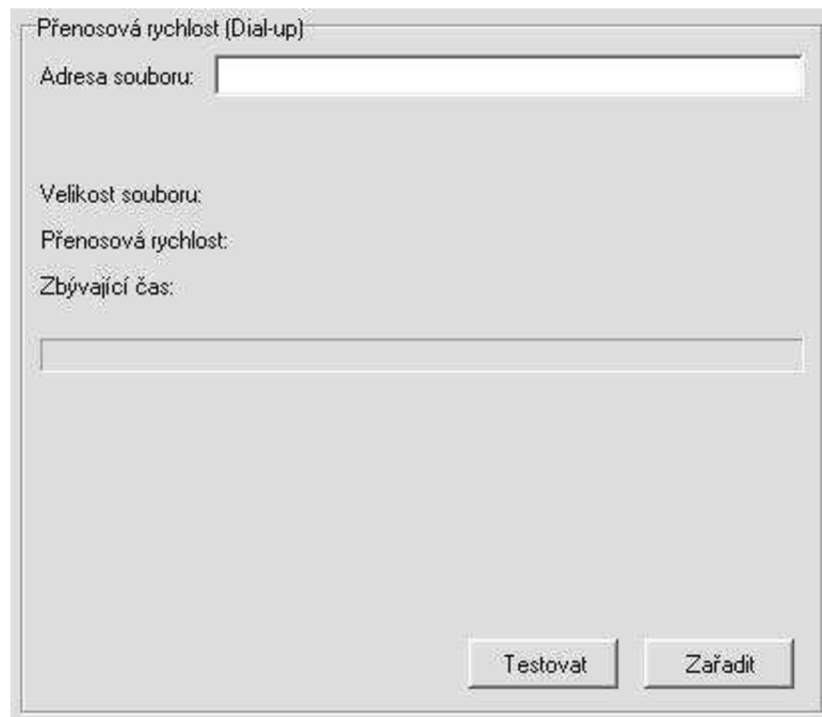
```

    start = clock(); // cas pri zacatku mereneho intervalu
}
if (((stop-start)/(double)CLK_TCK) >= 1){
    prenos = (stav2-stav1); // prenosova rychlost
    prumpren[0] += prenos; // ulozeni prenosove rychlosti pro
vypocet prumeru
    prumpren[1] += 1; // ulozeni cyklu pro vypocet prumeru
    stav1 = Readed;
    start = clock();
}

```

Výsledek měření je zapsán do textového souboru, který obsahuje standardní hlavičku a poté zápis obsahující adresu souboru, dobu stahování, počet přenesených bitů a průměrnou přenosovou rychlost.

Při volbě tlačítka Zařadit (BitBtn\_DZar) se test zařadí do sledu testování tak, že za zkratkou testu je obsažena adresa souboru.



Obr. 5.11: Test - Přenosová rychlost (Dial-up)

### 5.3.6 Vlastní test

Posledním testem, který je možné zvolit je Vlastní test. Jedná se o možnost postupně posílat AT příkazy mobilnímu zařízení.

AT příkazy je možné zvolit v databázi AT příkazů, které je na prvním místě v seznamu databází v záložce Databáze. Po výběru AT příkazu ze seznamu je možné nastavit jeho parametry. Po kliknutí na tlačítko Zadat v záložce Databáze se AT příkaz přidá do seznamu příkazů (ListBox\_Prik), které budou odesílány mobilnímu zařízení ve vlastním testu.

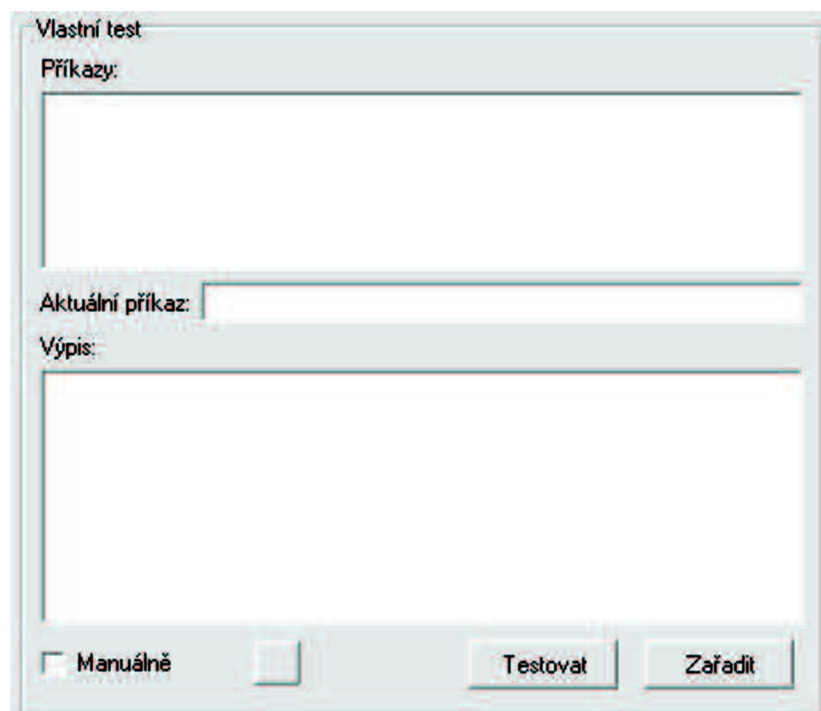
Posílat AT příkazy lze buď automaticky nebo manuálně. Pokud není zaškrtnuto políčko Manuálně (CheckBox\_VI), posílají se AT příkazy mobilnímu zařízení

automaticky po sobě po volbě tlačítka Testovat (BitBtn\_VTst) v krátkém intervalu bez možnosti zásahu. Reakce se pak vypisuje ve výpisu (Memo\_Vypis).

Pokud je zaškrtnuto políčko Manuálně, budou se AT příkazy postupně posílat mobilnímu zařízení jednotlivě po volbě pomocného tlačítka (SpeedButton\_Man). Tlačítko Testovat je ve chvíli zaškrtnutí políčka Manuálně neaktivní a naopak, viz zdrojový kód.

```
if (CheckBox_Vl->Checked == False){
    for (int i = 0; i <= ListBox_Prik->Items->Count; i++) {
        Edit_VPrik->Text = ListBox_Prik->Items->Strings[0];
        ListBox_Prik->Items->Delete(0);
        ComPort->WriteStr(Edit_VPrik->Text);
        wait(0.2);
        ComPort->ReadStr(Odp,100);
        Memo_Vypis->Text = Odp;
    }
}
else {
    Memo_Vypis->Lines->Add("Je zapnut manuální režim.");
}
```

Při kliknutí na tlačítko Zařadit (BitBtn\_VZar) se test zařadí do sledu testování tak, že za zkratkou testu jsou uvedeny AT příkazy, které se pak vysílají automaticky do mobilního zařízení.



Obr. 5.12: Test - Vlastní test

## 5.4 Chyby programu

Ve verzi, v které se nyní program nachází, bylo nalezeno několik chyb, které je nutné ošetřit pro plnou funkčnost programu. Jedná se hlavně o buffer sériové komunikace a alokování paměti při deklarování pomocných proměnných.

Při sériové komunikaci dochází nepravidelně k přeplnění vstupního a výstupního bufferu, což vede k hrubé chybě programu a jeho následnému ukončení. Tento problém byl řešen vymazáním bufferu hned po přijetí příchozích dat a stejně tak hned po odeslání dat. Toto řešení ale nepomohlo a plnění bufferu a v některých procedurách náhodné a nekorektní. Chyba může být způsobena také špatnou spoluprací doplňku ComPort Library a zbytku programu. Další možnost řešení je upravit vkládání a načítání dat z bufferu pomocí pomocné komponenty nebo výměna komponenty za jinou, které zprostředkovává komunikaci se sériovým portem.

Další významná chyba, která se objevuje při běhu programu je plnění proměnných náhodnými daty při běhu programu. V pomocných proměnných, které jsou korektně vytvořeny a inicializovány na hodnotu 0 se objevují data, která jsou požívána v jiných pomocných proměnných nebo byla používána a to i v případě, že proměnné spolu nesouvisí. Tento jev se objevuje zejména a deklarování pole znaků char. Pro řešení tohoto problému by bylo vhodné alokovat vždy část paměti a při každé změně ji přealokovat, viz následující příklad.

Nynější deklarace:

```
char *promenna;  
free(promenna);
```

Změna na deklaraci:

```
char *promenna;  
promenna = (char*)malloc(velikost*sizeof(char));  
promenna = (char*)realloc(pocet, velikost*sizeof(char));  
free(promenna);
```

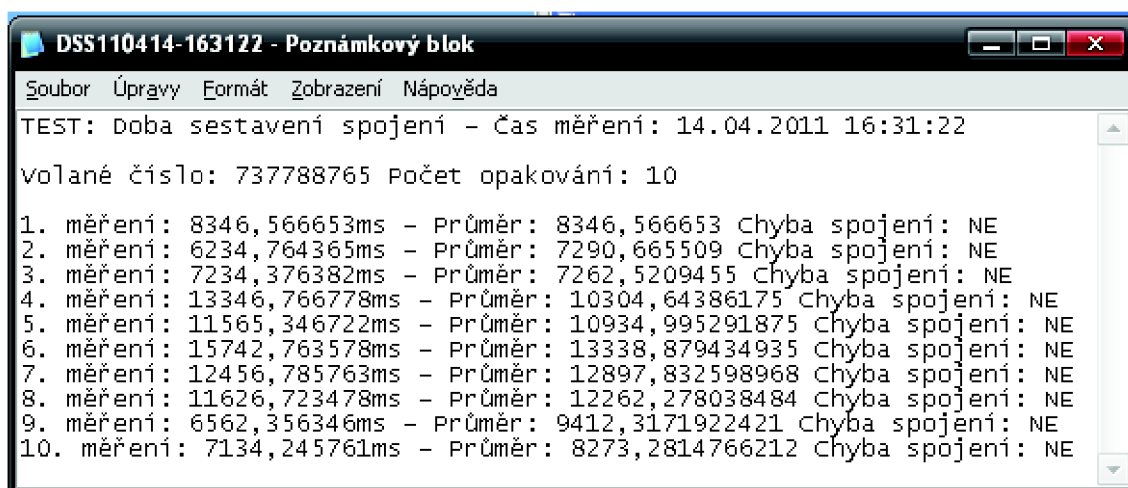
Při běhu programu se vyskytují i další chyby. Ty ale nezpůsobí kritické ukončení aplikace. Při jejich výskytu se však objevují nesmyslná data. Jedná se zejména o ošetření událostí, kterých se dopustí nepozorný uživatel, jako je například vyplnění písmen do pole určeného pouze pro číslice apod.

## 6 NAMĚŘENÉ A ZÍSKANÉ HODNOTY

Při vytváření jednotlivých testů byly naměřeny i příslušné hodnoty jednotlivých parametrů. Hodnoty jsou vyčteny z textových souborů, které byly vytvořeny jednotlivými testy.

### 6.1 Test DSS

Doba sestavení spojení byla poměrně proměnlivá. Důvodem může být úroveň signálu, úroveň rušení nebo i vytížení sítě v danou chvíli, či typ BTS, ke které bylo mobilní zařízení připojeno. Hodnoty jsou patrné z obr. 6.1, kterým je výpis textového souboru k danému testu.



The screenshot shows a Notepad window titled "DSS110414-163122 - Poznámkový blok". The text inside the window is as follows:

```
Soubor Úpravy Formát Zobrazení Nápověda
TEST: Doba sestavení spojení - Čas měření: 14.04.2011 16:31:22
Volané číslo: 737788765 Počet opakování: 10
1. měření: 8346,566653ms - Průměr: 8346,566653 Chyba spojení: NE
2. měření: 6234,764365ms - Průměr: 7290,665509 Chyba spojení: NE
3. měření: 7234,376382ms - Průměr: 7262,5209455 Chyba spojení: NE
4. měření: 13346,766778ms - Průměr: 10304,64386175 Chyba spojení: NE
5. měření: 11565,346722ms - Průměr: 10934,995291875 Chyba spojení: NE
6. měření: 15742,763578ms - Průměr: 13338,879434935 Chyba spojení: NE
7. měření: 12456,785763ms - Průměr: 12897,832598968 Chyba spojení: NE
8. měření: 11626,723478ms - Průměr: 12262,278038484 Chyba spojení: NE
9. měření: 6562,356346ms - Průměr: 9412,3171922421 Chyba spojení: NE
10. měření: 7134,245761ms - Průměr: 8273,2814766212 Chyba spojení: NE
```

Obr. 6.1: Výstup testu DSS

Pro lepší analýzu sítě a porovnání naměřených hodnot by bylo vhodné mít několik desítek sad naměřených hodnot.

### 6.2 Test DOS

Podobně jako u předchozího testu má na přenos dat vliv několik faktorů, jako jsou vytížení sítě, úroveň signálu apod. Přesto však doba odesílání SMS a následného přijetí vyznění byla konstantní. Hodnoty jsou vyobrazeny na obr. 6.2, kterým je výpis textového souboru k danému testu.

```
DOS110422-124142 - Poznámkový blok
Soubor Úpravy Formát Zobrazení Nápověda
TEST: Doba odeslání SMS - Čas měření: 22.04.2011 12:41:42
volané číslo: 737788765 Počet opakování: 5
Testovací zprava.
1. měření: 10324,54356ms - Průměr: 10324,54356 Chyba spojení: NE
2. měření: 9998,34553ms - Průměr: 10161,44454 Chyba spojení: NE
3. měření: 9999,54546ms - Průměr: 10080,49500 Chyba spojení: NE
4. měření: 10345,46576ms - Průměr: 10212,98038 Chyba spojení: NE
5. měření: 10245,67327ms - Průměr: 10229,32682 Chyba spojení: NE
```

Obr. 6.2: Výstup testu DOS

Pro lepší analýzu sítě a porovnání naměřených hodnot by bylo vhodné mít několik desítek sad naměřených hodnot.

### 6.3 Test PIN

Test PING měl poměrně různorodé hodnoty. To může být způsobeno řadou faktorů, ať již zmíněnými v předchozích podkapitolách, tak jiných, kterými mohly být aktuální vytížení serveru nebo mobilního zařízení, viz obr. 6.2, kterým je výpis textového souboru k danému testu.

```
PIN110520-060149 - Poznámkový blok
Soubor Úpravy Formát Zobrazení Nápověda
TEST: PING - Čas měření: 15.04.2011 16:01:49
Host: www.seznam.cz Počet opakování: 5
Příkaz PING na www.seznam.cz [77.75.72.3] s délkou 56 bajtů:
Odesílám na 77.75.72.3 (77.75.72.3): bajty=56 TTL=64
Odpověď od 77.75.72.3 (77.75.72.3): bajty=56 čas=14ms TTL=64
Odesílám na 77.75.72.3 (77.75.72.3): bajty=56 TTL=64
Odpověď od 77.75.72.3 (77.75.72.3): bajty=56 čas=33ms TTL=64
Odesílám na 77.75.72.3 (77.75.72.3): bajty=56 TTL=64
Odpověď od 77.75.72.3 (77.75.72.3): bajty=56 čas=17ms TTL=64
Odesílám na 77.75.72.3 (77.75.72.3): bajty=56 TTL=64
Odpověď od 77.75.72.3 (77.75.72.3): bajty=56 čas=34ms TTL=64
Odesílám na 77.75.72.3 (77.75.72.3): bajty=56 TTL=64
Odpověď od 77.75.72.3 (77.75.72.3): bajty=56 čas=19ms TTL=64

Statistika ping pro 77.75.72.3:
Pakety: Odeslané = 5, Přijaté = 5, Ztracené = 0 (ztráta 0%),
Přibližná doba do přijetí odezvy v milisekundách:
Minimum = 17ms, Maximum = 34ms, Průměr = 23ms|
```

Obr. 6.3: Výstup testu PIN

K porovnání je vhodné např. uvést hodnoty měřené na PC:

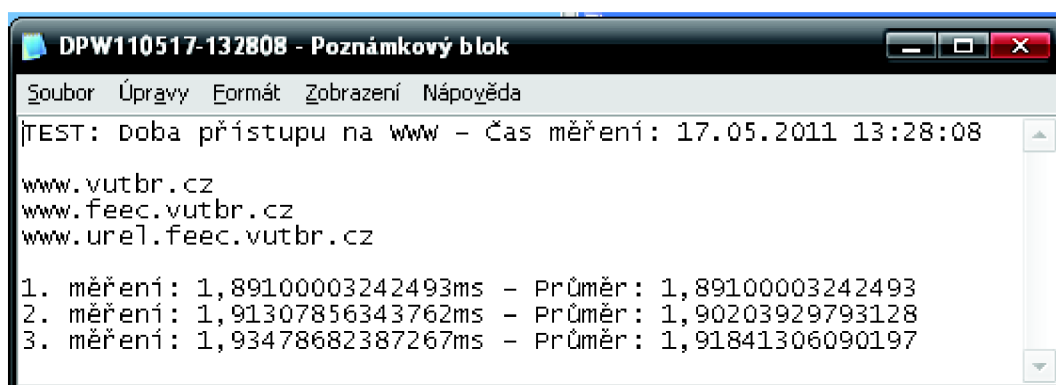
Příkaz PING na www.seznam.cz [77.75.76.3] s délkou 32 bajtů:

```
Odpověď od 77.75.76.3: bajty=32 čas=14ms TTL=250
Odpověď od 77.75.76.3: bajty=32 čas=15ms TTL=250
Odpověď od 77.75.76.3: bajty=32 čas=14ms TTL=250
Odpověď od 77.75.76.3: bajty=32 čas=15ms TTL=250
```

```
Statistika ping pro 77.75.76.3:
Pakety: Odeslané = 4, Přijaté = 4, Ztracené = 0 (ztráta 0%),
Přibližná doba do přijetí odezvy v milisekundách:
  Minimum = 14ms, Maximum = 15ms, Průměr = 14ms
```

## 6.4 Test DPW

Doba načítání internetových stránek je poměrně rozličná, viz obr 6.4, kterým je výpis textového souboru k danému testu. Při prvním spuštění je načítání vždy prodlouženo o dobu spuštění Internet Exploreru. Poté závisí doba na obsahu stránek, přenosové cestě apod. Testování stránek by zřejmě bylo vhodně rozdělit do několika kategorií, které by se rozlišovaly podle obsahu stránek, tedy jaké prvky obsahují a jak jsou náročné pro zobrazení.



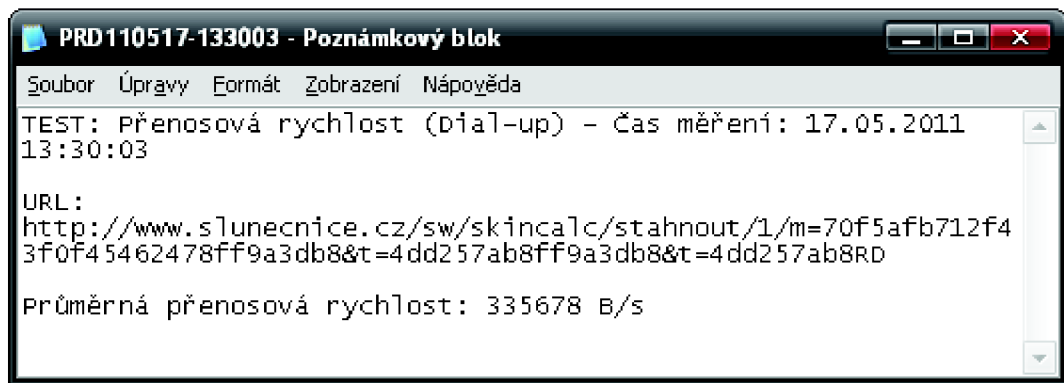
```
DPW110517-132808 - Poznámkový blok
Soubor Úpravy Formát Zobrazení Nápověda
TEST: Doba přístupu na www - Čas měření: 17.05.2011 13:28:08
www.vutbr.cz
www.feec.vutbr.cz
www.urel.feec.vutbr.cz
1. měření: 1,89100003242493ms - Průměr: 1,89100003242493
2. měření: 1,91307856343762ms - Průměr: 1,90203929793128
3. měření: 1,93478682387267ms - Průměr: 1,91841306090197
```

Obr. 6.4: Výstup testu DPW

## 6.5 Test PRD

Testování přenosové rychlosti připojení Dial-up testuje síť z hlediska stahování jednoho většího souboru, na rozdíl od testu DPW, kde se stahuje mnoho malých souborů. Přenosová rychlost je pak poměrně konstantní, pokud jsou zajištěny dobré podmínky příjmu, viz 6.5, kterým je výpis textového souboru k danému testu.





Obr. 6.5: Výstup testu PRD

Pro lepší analýzu sítě a porovnání naměřených hodnot by bylo vhodné stahování několikrát opakovat a výsledky spojit do jednoho grafu.

## 7 ZÁVĚR

Předmětem diplomové práce na téma „Měření kvality a dostupnosti GSM sítě“ byla analýza měřících technik a jejich realizace, návrh vlastní testů pro měření sítě z pohledu běžného uživatele, rozbor možností pro realizaci a vypracování měřícího systému, který by splňoval daná kritéria.

Po prvotním rozboru měřících technik, možností měření a měřících systémů bylo potřebné definovat požadavky na měřící systém. Definování požadavků, které by měl měřící systém splňovat bylo poměrně komplikované, protože komunikace s původním zadavatelem práce byla nepřímá a absorbovala příliš mnoho času, který byl vyhrazený na práci dalšího postupu. Z tohoto důvodu bylo přistoupeno na vlastní řešení práce vycházející z dosud známých skutečností.

Původní verze návrhu v prostředí Matlab byla poměrně složitá. Matlab sice sám o sobě nabízel některé prostředky vhodné pro řešení diplomové práce, ale nemohl nahradit řadu komponent, kterými disponuje např. Borland C++ Builder 6.0. Tato cesta se tedy ukázala jako nevhodná a proto byly vypracovány další možnosti řešení práce, které jsou uvedeny v kapitole 4. Jako výsledné řešení bylo zvoleno naprogramování ovládací aplikace v prostředí Borland C++ Builder 6.0, přičemž komunikaci se sítí GSM bude zprostředkovávat mobilní zařízení spojené s počítačem přes rozhraní Bluetooth.

Postupným řešením diplomové práce se objevilo několik omezení, která znemožňují nebo upravují původní záměry a mění tak i zadané požadavky na vytvářený program. Jedním z omezení je nemožnost přímo zasahovat do nastavení mobilní stanice a tak měnit používané technologie pro datový přenos nebo měnit BTS, na kterou je mobilní stanice připojena. Dalším omezením je zkrácení parametrů měření, protože není možno data získávat přímo z mobilní stanice, ale zprostředkovaně. Kromě zmíněných problémů v řešení je zde také omezení v počtu připojených modemů k počítači a komunikace pomocí nich s internetem, zejména u operačního systému Microsoft Windows XP. Některá tato omezení je možné řešit do určité míry pomocí AT příkazů nebo přímým odkazováním na daný port.

Výsledkem diplomové práce je vlastní aplikace, programovaná v posledním semestru, určená pro operační systém MS Windows XP nebo NT nazvaná Mobiler. Již od počátku byla vytvářena tak, aby mohla komunikovat se všemi typy mobilních zařízení, která jsou připojena přes rozhraní Bluetooth k počítači, na kterém pracuje Mobiler. Výsledný program je schopen se spojit přes port COM nebo pomocí Dial-up s libovolným zařízením, které však musí být nainstalováno v operačním systému a spárováno s ním. Pokud je vše provedeno, je možné se zařízením navázat komunikaci a provádět naprogramované testy.

Síť je možno testovat pomocí předdefinovaných testů nebo vlastního. Jedná se o dobu sestavení spojení, dobu odesílání SMS zpráv, test PING, dobu načítání internetové stránky, přenosovou rychlost stahovaného souboru. Výsledkem testů jsou buď textové soubory nebo databázové tabulky určené k dalšímu zpracování, jako je export nebo tvorba grafu.

Výhodou programu je možnost vytvářet vlastní sled testů, který nepotřebuje

obsahu a pracuje zcela samostatně.

Vzhledem ke krátké době, po kterou byla aplikace vyvíjena, bylo objeveno a mnoho nedostatků, které se stále odstraňují. Podstatným je občasná ztráta komunikace, kdy dojde k přeplnění buferů pro datový přenos a aplikace havaruje. Dále pak práce s pamětí. V paměti nových znakových proměnných se objevují staré proměnné nebo jiné, které jsou zastoupeny libovolným počtem znaků.

Protože testy byly naprogramovány, aby fungovaly principiálně správně, je zde velký prostor pro vylezení některých chyb, jako je špatné načítání hlavičky u některých stahovaných souborů nebo špatná práce s daty při nedodržení vstupních podmínek.

U programu by bylo vhodné ošetřit uživatelské události a minimalizovat tak možnost špatné funkce při chybě obsluhy programu. Tím by se stal program více intuitivní a uživatelsky přívětivější. K uživatelské přívětivosti by přispělo i odprogramování několika funkcí, kterou je např. posouvání položek v seznamu naplánovaných testů apod.

Z důvodu posunutí práce na programu na poslední část stanoveného času nebylo změřeno dostatečné množství vzorků pro adekvátní porovnávání a výsledkem změřených hodnot jsou pouze výpisy testů.

Přes všechny nedostatky, kterými program trpí, je stanoven základ, který je možno vylepšovat a odlazovat. Po odlazení by bylo vhodné optimalizovat zdrojový kód a odstranit tak opakující se sekvence. Dále by pak bylo vhodné doplnit podporu pro připojení dalších mobilních zařízení, aby bylo možno testovat více operátorů zároveň a srovnávat je tak. Kromě rozšířené podpory by bylo vhodné doplnit program o další testy a testovat pak např. vzájemnou ICQ komunikaci mezi dvěma přístroji.

Pokud se porovnají původní cíle řešení vytvářeného programu s komerčním řešením poslední verze QVoice, není zde příliš velký rozdíl ve smyslu měření datových služeb a jejich parametrů. Protože systém QVoice umožňuje nadefinování vlastních scénářů měření a také vytvoření vlastních výstupních závislostí měření. Výsledek této práce se tedy nemůže příliš přiblížit systému QVoice, neboť se zaměřuje spíše pouze na ovládací program a nikoliv i na hardwarové řešení.

# LITERATURA

- [1] Wikipedia. *Wikipedia* [online]. 23.4.2010 [cit. 2010-04-23]. 3GPP. Dostupné z WWW: <http://en.wikipedia.org/wiki/3GPP>.
- [2] 3GPP. *3GPP* [online]. 2008-09 [cit. 2010-04-23]. 3gpp.org. Dostupné z WWW: [http://www.3gpp.org/ftp/Information/WORK\\_PLAN/Description\\_Releases/Previous\\_versions/Rel-9\\_description\\_20080911.zip](http://www.3gpp.org/ftp/Information/WORK_PLAN/Description_Releases/Previous_versions/Rel-9_description_20080911.zip).
- [3] CELKOM. *CELKOM* [online]. Praha : 2009 [cit. 2010-04-23]. Princip fungování GSM sítě. Dostupné z WWW: <http://www.zesilovac-signalu.cz/cs/princip-fungovani-gsm-site>.
- [4] RAK, Milan. *Blokové schéma mobilního telefonu GSM*. 3 s.
- [5] *Wikimedia Česká republika* [online]. 1.3.2010 [cit. 2010-04-25]. General Packet Radio Service. Dostupné z WWW: <http://cs.wikipedia.org/wiki/GPRS>.
- [6] *DenMas BROTO* [online]. 28.5.2008 [cit. 2010-04-25]. GPRS Network Architecture. Dostupné z WWW: <http://www.denmasbroto.com/?pilih=news&mod=yes&aksi=lihat&id=5>.
- [7] *Wikimedia Česká republika* [online]. 24.1.2010 [cit. 2010-04-26]. EGPRS. Dostupné z WWW: <http://cs.wikipedia.org/wiki/EGPRS>.
- [8] CDMA. *CDMA.cz* [online]. 2004 - 2010 [cit. 2010-04-26]. Jak funguje CDMA. Dostupné z WWW: [www.cdma.cz/jak-cdma-funguje](http://www.cdma.cz/jak-cdma-funguje).
- [9] KREHER, Ralf, et al. *UMTS UTRAN Signaling*. U.S. : Tektronix, 2003. 23 s.
- [10] BEŠŤÁK, R.; PRAVDA, I. *Access server* [online]. 30.12.2005 [cit. 2010-05-07]. Sítě UMTS. Dostupné z WWW: <http://access.feld.cvut.cz/view.php?cislocianku=2005113001>.
- [11] ČEPIČKA, David. *PCWorld* [online]. 10.02.2009 [cit. 2011-05-20]. Základy technologie Bluetooth: původ a rozsah funkcí. Dostupné z WWW: <http://pcworld.cz/hardware/Zaklady-technologie-Bluetooth-puvod-a-rozsah-funkci-6635>.
- [12] ČEPIČKA, David. *PCWorld* [online]. 18.02.2009 [cit. 2011-05-20]. Základy technologie Bluetooth: komunikace a zabezpečení. Dostupné z WWW: <http://pcworld.cz/hardware/Zaklady-technologie-Bluetooth-komunikace-a-zabezpeceni-6636>.
- [13] Fichier:Bluetooth logo.png. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 11.12.2005, last modified on 28.8.2008 [cit. 2011-05-18]. Dostupné z WWW: [http://fr.wikipedia.org/wiki/Fichier:Bluetooth\\_logo.png](http://fr.wikipedia.org/wiki/Fichier:Bluetooth_logo.png).
- [14] HANUS, Stanislav; FENCL, Josef; ŠTENCEL, Vít. *Rádiové a mobilní komunikace II*. Brno : MJ servis spol. s r.o., 2010. Měření v mobilní síti GSM, s. 65-120.
- [15] *Ascom* [online]. 2010 [cit. 2011-01-01]. Dostupné z WWW: <http://www.ascom.com/en/index.htm>.
- [16] CyberSoft s.r.o. *Obchod.hw.cz* [online]. 2011 [cit. 2011-01-01]. GSM Industry. Dostupné z WWW: <http://obchod.hw.cz/?cls=spresenttrees&strid=35>.

# SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ADC	Administrative Centre
BG	Border Gateway
BSC	Base Station Controller
BSS	Base Station Subsystem
BTS	Base Transceiver Station
CDMA	Code Division Multiple Access
CG	Charging Gateway
COM	Component Object Model
CSD	Circuit Switched Data
DTM	Dual Transfer Mode
EDGE	Enhanced Data rates for Global Evolution Enhanced Data rates for GSM Evolution
EGPRS	Enhanced GPRS
EIR	Equipment Identity Register
EV-DO	Evolution Data Only
FAQ	Frequently Asked Questions
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communication, Groupe Spécial Mobile
GMSK	Gaussian Minimum Shift Keying
GNU	GNU's Not Unix!
GTP	GPRS Tunelling Protocol
HLR	Home Location Register
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
ISDN	Integrated Services Digital Network
IWF	Inter-Working Functionality
J2ME	Java 2 Micro Edition

LAI	Location Area Identity
LIG	Legal Interception Gateway
MS	Mobilní Stanice (mobilní telefon)
MSC	Mobile Switching Centre
MSISDN	Mobile Subscriber ISDN Number
NMC	Network Management Centre
NSS	Network Switching Subsystem
OMC	Operational and Maintenance Centre
OSS	Operational and Support Subsystem
PIN	Personal Identification Number
PUK	Personal Unblocking Key
RNC	Radio Network Controller
SE	Sony Ericsson
SIM	Subscriber Identification Module
SGSN	Serving GPRS Support Node
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TMSI	Temporary Mobile Subscriber Identity
UMTS	Universal Mobile Telecommunication System
UTRAN	UMTS Terrestrial Radio Access Network
USRAN	UMTS Satellite Radio Access Network
VaC	Authentication Centre
VLR	Visitor Location Register
WiMAX	Worldwide Interoperability for Microwave Access
WPAN	Wireless Personal Area Networks
BSIG	Bluetooth Special Interest Group
AFHSS	Advance Frequency Hopping Spread Spektrum
SCO	Synchronous Connection Oriented Link
EDR	Enhanced Data Rate
802.11 PAL	802.11 Protocol Adaption Layer
MAC/PHY	Media Access Control/Physical
EPC	Enhanced Power Kontrol
AES-128	Advanced Encryption Standard 128-bitový
ISM	Industrial, Scientific and Medical
FHSS	Frequency Hopping Spread Spektrum

BD_ADDR	BlueTooth Device Address
SDP	Service Discovery Protocol
TDD	Time Division Duplex
GFSK	Gaussian Frequency Shift Keying
AM_ADDR	Active Member Address
SCO	Synchronous Connection Oriented Link
ACL	Asynchronous Connectionless Link
DM	Data Medium Rate
DH	Data High Rate
FEC	Forward Error Correction
HV	High Quality Voice

# SEZNAM PŘÍLOH

<b>A</b>	<b>Seznam AT příkazů</b>	<b>61</b>
A.1	GSM 07.05 .....	61
A.2	GSM 07.07 .....	62
<b>B</b>	<b>Zdrojový kód programu v 1.1.0</b>	<b>66</b>
B.1	Main.h .....	66
B.2	Main.cpp .....	72



# A SEZNAM AT PŘÍKAZŮ

## A.1 GSM 07.05

### Textový mód

#### Obecné konfigurační příkazy

+CSMS	Výběr služby zpráv
+CPMS	Preferované ukládání zpráv
+CMGF	Formát zprávy
+CESP	Zadání SMS Block Mode protokolu
+CMS ERROR	Chyba výsledného kódu služby zpráv

#### Konfigurační příkazy pro zprávy

+CSCA	Adresa servisního střediska
+CSMP	Nastavení parametrů textového módu
+CSDH	Zobrazení parametrů textového módu
+CSCB	Výběr typu buněk broadcast zprávy
+CSAS	Uložení nastavení
+CRES	Obnovení nastavení

#### Příkazy pro přijímání a čtení zpráv

+CNMI	Indikátor nové zprávy
+CMGL	Seznam zpráv
+CMGR	Čtení zprávy
+CNMA	Nová zpráva – poděkování ME/TA

#### Příkazy pro posílání a čtení zpráv

+CMGS	Odeslání zprávy
+CMSS	Odeslání zprávy z úložiště
+CMGW	Uložení zprávy do paměti
+CMGD	Smazání zprávy
+CMGC	Příkaz k odeslání
+CMMS	Více zpráv k odeslání

### **PDU mód**

+CMGL	Seznam zpráv
+CMGR	Čtení zprávy
+CMGS	Odeslání zprávy
+CMGW	Uložení zprávy do paměti
+CMGC	Příkaz k odeslání
+CNMA	Nová zpráva – poděkování ME/TA
+CMSS	Odeslání zprávy z úložiště

## **A.2 GSM 07.07**

### **Obecné příkazy**

S3	Ukončovací znak příkazového řádku (povinné nastavení)
S4	Odpověď formátování znaků
S5	Editovací znak příkazového řádku
E	Příkaz echo – povolení nebo zakázání
Q	Potlačení kódu výsledku
V	Formát odpovědi terminálu
X	Definování formátu kódu výsledku připojení
&C	Detekce a odpověď linky příj. signálu ze vzdáleného konce
&D	Def. změnu odpovědi při změně podm. z ON/OFF během přenosu
+IPR	Pevná rychlost přenosu dat (primárně automatická)
+ICF	Charakter rámování
+IFC	Místní řízení toku dat
+ILRR	Zda použít lokální rychlost přenosu dat pomocí mezivýsledku
+CGMI	Dotaz na identifikaci výrobce
+CGMM	Dotaz na identifikaci modelu zařízení
+CGMR	Dotaz na vydání zařízení
+CGSN	Dotaz na sériové číslo produktu
+CSCS	Výběr nastavení znaků MS
+CIMI	Dotaz na international mobile subscriber identity
+CMUX	Multiplexní mód
+WS46	Výběr bezdrátové sítě

### **Příkazy pro řízení volání a metody**

+CSTA	Výběr typů adres
+CMOD	Mód volání

+CHUP	Ukončení hovoru
+CBST	Volba typu nosiče služby
+CRLP	Protokol rádiového spojení
+CR	Služba hlášení ovládnání
+CEER	Rozšířené ohlášení chyb
+CRC	Výsledný kód sítě
+CHSD	HSCSD parametry zařízení
+CHST	HSCSD transparentní konfigurace volání
+CHSN	HSCSD netransparentní konfigurace volání
+CHSC	HSCSD nynější parametry volání
+CHSR	HSCSD hlášení parametrů
+CHSU	HSCSD automatické zahájení modernizace uživatele
+CSNS	Jednotné schéma číslování
+CVHU	Zavěšení volání
+CV120	Protokol adaptace rychlosti
D	Příkaz k volání čísla
A	Přijmutí příchozího hovoru
H	Ukončení hovoru
O	On-line stav příkazového režimu (nutná hodnota 0)
S0	Nastavuje počet zvonění před automatickým přijmutím, 0 vyp.
S7	Počet sekund do vytvoření hovoru, jinak proces ukončí
S8	Počet sekund čekání před ukončení příkazu ATD při zadání „ , “
S10	Počet desetin sekundy před odpojením
+DS	Kompresi dat
+DR	Určování stavu z mezivýsledku (pro kompresi dat)

#### **Příkazy související s Network service**

+CNUM	Číslo účastníka
+CREG	Registrace sítě
+COPS	Výběr operátora
+CLCK	Zámek zařízení
+CPWD	Změna hesla
+CLIP	Identifikace volající linky
+CLIR	Identifikace omezení volajícího
+COLP	Prezentace připojení identifikační linky
+CCUG	Uzavřená skupina uživatelů

+CCFC	Počet a podmínky přesměrování volání
+CCWA	Čekání hovoru
+CHLD	Výzva související doplňkové služby
+CTFR	Deformace volání
+CUSD	Nestrukturovaná data doplňkové služby
+CAOC	Informace o poplatku
+CSSN	Oznámení doplňkové služby
+CLCC	Seznam aktuálních výzev
+CPOL	Seznam preferovaných operátorů
+COPN	Čtení jmen operátorů

#### **Příkazy kontroly a statusu mobilního zařízení**

+CPAS	Status aktivity telefonu
+CFUN	Nastavení funkcionality telefonu
+CPIN	Vložení PINu
+CBC	Nabíjení baterie
+CSQ	Kvalita signálu
+CMEC	Ovládací mód mobilního zařízení
+CKPD	Ovládání klávesnice
+CDIS	Ovládání displeje
+CIND	Indikátor ovládání
+CMER	Hlášení událostí mobilního zařízení
+CPBS	Výběr paměťového prostoru pro adresář
+CPBR	Čtení záznamů adresáře
+CPBF	Hledání záznamu v adresáři
+CPBW	Zápis záznamu do adresáře
+CCLK	Hodiny
+CALA	Alarm
+CSIM	Druhový přístup k SIM
+CRSM	Omezený přístup k SIM
+CSCC	Příkaz bezpečného ovládání
+CALM	Nastavení zvuku poplachu
+CRSL	Úroveň vyzvánění
+CVIB	Vibrační mód
+CLVL	Úroveň hlasitosti reproduktoru
+CMUT	Ovládání mute

+CACM	Počítadlo volání
+CAMM	Maximum počítadla volání
+CPUC	Cena za jednotku a tabulka měny
+CCWE	Výzva počítadla maximální události
+CPWC	Výkonová třída
+CLAN	Nastavení jazyku
+CLAE	Jazykové události
+CSGT	Nastavení textu pozdravu
+CSVM	Nastavení čísla hlasové schránky
+CRMP	Melodie vyzvánění
+CMAR	Hlavní reset
+CLAC	Seznam možných AT příkazů

#### **Errorry mobilního zařízení**

+CMEE	Hlášení chyb mobilního zařízení
+CME ERROR	Výsledný kód chyb mobilního zařízení

#### **Příkazy pro GPRS**

+CGDCONT	Definování PDP kontextu
+CGQREQ	Profil kvality služeb - požádaných
+CGQMIN	Profil kvality služeb – akceptovatelné minimum
+CGATT	GPRS připojení nebo odpojení
+CGACT	Aktivování nebo deaktivování PDP kontextu
+CGDATA	Zadání stavu dat
+CGCLOSP	Konfigurace parametrů místního Octet Stream PAD
+CGPADDR	Ukázání PDP adresy
+CGAUTO	Aktivace automatické odpovědi na dotaz sítě PDP kontextu
+CGANS	Aktivace manuální odpovědi na dotaz sítě PDP kontextu
+CGCLASS	GPRS třída mobilní stanice
+CGCLPAD	Konfigurace parametrů místní triple-X PAD
+CGEREP	GPRS oznámení událostí
+CGREG	GPRS status registrace sítě
+CGSMS	Výběr služby pro SMS služby

# B ZDROJOVÝ KÓD PROGRAMU V 1.1.0

## B.1 Main.h

```
//-----  
-----  
  
#ifndef MainH  
#define MainH  
//-----  
-----  
  
#include <Classes.hpp>  
#include <Controls.hpp>  
#include <StdCtrls.hpp>  
#include <Forms.hpp>  
#include <ComCtrls.hpp>  
#include <Menus.hpp>  
#include <ToolWin.hpp>  
#include <ExtCtrls.hpp>  
#include <Graphics.hpp>  
#include <Buttons.hpp>  
#include "CPort.hpp"  
#include "CPortCtl.hpp"  
#include <SysUtils.hpp>  
#include <Messages.hpp>  
#include <Windows.hpp>  
#include <SysInit.hpp>  
#include <System.hpp>  
#include <Dialogs.hpp>  
#include <DB.hpp>  
#include <DBCtrls.hpp>  
#include <DBGrids.hpp>  
#include <DbChart.hpp>  
#include <DBTables.hpp>  
#include <Grids.hpp>  
#include <Chart.hpp>  
#include <TeEngine.hpp>  
#include <TeeProcs.hpp>  
#include "RasManager.hpp"  
#include "Ping.hpp"  
#include "HttpProt.hpp"  
#include "SHDocVw_OCX.h"  
#include <OleCtrls.hpp>  
#include "HTTPGet.hpp"  
//-----  
-----  
  
class TForm1 : public TForm  
{  
    __published: // IDE-managed Components  
        TStatusBar *StatusBar;  
        TMainMenu *MainMenu;  
        TToolBar *ToolBar;  
        TToolButton *ToolButton2;  
        TToolButton *ToolButton3;  
        TPageControl *PageControl;  
        TTabSheet *TabSheet_Pripoj;  
};
```

```

TTabSheet *TabSheet_Graf;
TTabSheet *TabSheet_Data;
TTabSheet *TabSheet_O;
TSplitter *Splitter;
TImage *Image_VUT;
TImage *Image_FEKT;
TLabel *Label_VUT;
TLabel *Label_FEKT2;
TLabel *Label_FEKT1;
TLabel *Label_NazevProg;
TLabel *Label_Verze;
TLabel *Label_Dipl;
TLabel *Label_Tema;
TLabel *Label_Ved;
TLabel *Label_Autor;
TLabel *Label_Rok;
TLabel *Label_VUTAdr;
TLabel *Label_FEKTAdr;
TLabel *Label_Moznosti;
TLabel *Label_PripojeZar;
TLabel *Label_Grafy;
TLabel *Label_Databaze;
TBitBtn *BitBtn_Nap;
TBitBtn *BitBtn_COM;
TGroupBox *GroupBox_Param;
TGroupBox *GroupBox_Zariz;
TLabel *Label_COM;
TLabel *Label_Kom;
TBitBtn *BitBtn_Refr;
TLabel *Label_DB;
TLabel *Label_Pa;
TLabel *Label_ST;
TMenuItem *Soubor1;
TMenuItem *Ukonit1;
TComPort *ComPort;
TComComboBox *ComComboBox_COM;
TComComboBox *ComComboBox_DB;
TComComboBox *ComComboBox_Pa;
TComComboBox *ComComboBox_SB;
TComComboBox *ComComboBox_TD;
TComComboBox *ComComboBox_Kom;
TLabel *Label_TD;
TLabel *Label_Vyr;
TEdit *Edit_CGMI;
TDBNavigator *DBNavigator;
TDataSource *DataSource;
TTable *Table;
TDBGrid *DBGrid;
TDBChart *DBChart;
TTabSheet *TabSheet_Test;
TLabel *Label_Model;
TLabel *Label_Serie;
TEdit *Edit_CGMM;
TEdit *Edit_CGMR;
TEdit *Edit_CGSN;
TEdit *Edit_CIMI;
TLabel *Label_SerC;
TLabel *Label_Iden;
TTabSheet *TabSheet_Plan;
TMonthCalendar *MonthCalendar;
TMemo *Memo_Ins;

```

```

TRasManager *RasManager;
TBitBtn *BitBtn_Dial;
TBitBtn *BitBtn_Hang;
TLabel *Label_LC;
TLabel *Label_Stav;
TLabel *Label_IDB;
TLabel *Label_Sils;
TEdit *Edit_CCLK;
TEdit *Edit_CBC;
TEdit *Edit_CSQ;
TLabel *Label_Oper;
TEdit *Edit_CREG;
TMemo *Memo_COPS;
TLabel *Label_Vyb;
TListBox *ListBox_Vyb;
TGroupBox *GroupBox_ParAT;
TLabel *Label_Exp;
TBitBtn *BitBtn_Exc;
TBitBtn *BitBtn_Wor;
TLabel *Label_P2;
TLabel *Label_P4;
TLabel *Label_P1;
TLabel *Label_P3;
TLabel *Label_P5;
TLabel *Label_P6;
TLabel *Label_P10;
TLabel *Label_P7;
TLabel *Label_P9;
TLabel *Label_P8;
TEdit *Edit_Par3;
TBitBtn *BitBtn_Zad;
TEdit *Edit_Par4;
TEdit *Edit_Par2;
TEdit *Edit_Par5;
TEdit *Edit_Par1;
TEdit *Edit_Par6;
TEdit *Edit_Par7;
TEdit *Edit_Par8;
TEdit *Edit_Par9;
TEdit *Edit_Par10;
TLabel *Label_Zap;
TLabel *Label_Dial;
TLabel *Label_VybD;
TLabel *Label_Graf;
TLabel *Label_X;
TLabel *Label_Y;
TComboBox *ComboBox_Graf;
TListBox *ListBox_VybD;
TGroupBox *GroupBox_Graf;
TBitBtn *BitBtn_ExpG;
TLabel *Label_PopGraf;
TLabel *Label_Nazev;
TLabel *Label_OsaX;
TLabel *Label_OsaY;
TEdit *Edit_Nazev;
TEdit *Edit_OsaX;
TEdit *Edit_OsaY;
TComboBox *ComboBox_OsaX;
TComboBox *ComboBox_OsaY;
TListBox *ListBox_Sez;
TListBox *ListBox_Tst;

```



```

TLabel *Label_Tst;
TMemo *Memo_Test;
TLabel *Label_Test;
TBitBtn *BitBtn_Novy;
TMemo *Memo_Testy;
TLabel *Label_Dnes;
TLabel *Label_DnesDat;
TLabel *Label_DnesCas;
TLabel *Label_Datum;
TLabel *Label_Cas;
TEdit *Edit_Datum;
TEdit *Edit_Cas;
TLabel *Label_StartTst;
TEdit *Edit_CesStarTes;
TSpeedButton *SpeedButton_StarTes;
TLabel *Label_Sled;
TEdit *Edit_Sled;
TSpeedButton *SpeedButton_Sled;
TLabel *Label_Testy;
TMemo *Memo_Nact;
TLabel *Label_Nact;
TBitBtn *BitBtn_Apl;
TGroupBox *GroupBox_Testy;
TBitBtn *BitBtn_Start;
TBitBtn *BitBtn_Stop;
TLabel *Label_Poc;
TEdit *Edit_Poc;
TBitBtn *BitBtn_Aply;
TGroupBox *GroupBox_SMS;
TGroupBox *GroupBox_PING;
TGroupBox *GroupBox_Vlastni;
TGroupBox *GroupBox_WWW;
TGroupBox *GroupBox_PrenosDial;
TBitBtn *BitBtn_DTest;
TBitBtn *BitBtn_DZar;
TBitBtn *BitBtnTstSMS;
TBitBtn *BitBtn_ZarSMS;
TLabel *Label_SMS;
TMemo *Memo_SMS;
TLabel *Label_PocZn;
TLabel *Label_PocZnn;
TLabel *Label_PocZp;
TLabel *Label_PocZpp;
TEdit *Edit_TelCis;
TLabel *Label_TelCis;
TLabel *Label_PocOpak;
TEdit *Edit_PocOpak;
TLabel *Label_Nahr;
TEdit *Edit_Nahr;
TSpeedButton *SpeedButton_Nahr;
TBitBtn *BitBtn_Nahr;
TLabel *Label_PripZarPop;
TLabel *Label_Testov;
TLabel *Label_PlanTest;
TLabel *Label_PripZarPopp;
TLabel *Label_TestovPop;
TLabel *Label_PlanTestPop;
TLabel *Label_TestovPopp;
TLabel *Label20;
TEdit *Edit4;
TLabel *Label21;

```

```

TLabel *Label_PlanTestPopp;
TLabel *Label_DatabazePop;
TLabel *Label_GrafyPop;
TGroupBox *GroupBox_Volani;
TLabel *Label_Tel;
TLabel *Label_Opak;
TLabel *Label_Pop;
TEdit *Edit_Tel;
TEdit *Edit_Opak;
TBitBtn *BitBtn_Tst;
TBitBtn *BitBtn_Zar;
TStringGrid *StringGrid_Ses;
TOpenDialog *OpenDialog;
TMemo *Memo_Nacten;
TBitBtn *BitBtn_Uloz;
TSaveDialog *SaveDialog;
TLabel *Label_PopSMS;
TStringGrid *StringGrid_SMS;
TEdit *Edit_PiHost;
TLabel *Label_PiHost;
TMemo *Memo_Pi;
TLabel *Label_PiVyp;
TBitBtn *BitBtn_PiZar;
TBitBtn *BitBtn_PiTst;
TLabel *Label_PiOpak;
TEdit *Edit_PiOpak;
TLabel *Label_PiTime;
TEdit *Edit_PiTime;
TLabel *Label_PiVel;
TEdit *Edit_PiVel;
TLabel *Label_PiOut;
TEdit *Edit_PiOut;
TPing *Ping;
TBitBtn *BitBtn_WZar;
TBitBtn *BitBtn_WTst;
TMemo *Memo_WSez;
TLabel *Label_SeZ;
TCppWebBrowser *CppWebBrowser;
TLabel *Label_WPrub;
TEdit *Edit_Pom;
TSpeedButton *SpeedButton_Pom;
TStringGrid *StringGrid_WWW;
TLabel *Label_DURL;
TEdit *Edit_DURL;
TLabel *Label_DOut;
TEdit *Edit_DOut;
TProgressBar *ProgressBar;
TLabel *Label_DVel;
TLabel *Label_DPren;
TLabel *Label_DTime;
THTTPGet *HTTPGet;
TLabel *Label_DVell;
TLabel *Label_DPrenn;
TLabel *Label_DTimm;
TLabel *Label_DErr;
TLabel *Label_DFull;
TLabel *Label_DStat;
TTimer *Timer;
TSaveDialog *SaveDialog_Dial;
TBitBtn *BitBtn_VTst;
TBitBtn *BitBtn_VZar;

```

```

TMemo *Memo_Vypis;
TLabel *Label_Vyp;
TLabel *Label_VPrik;
TCheckBox *CheckBox_Vl;
TSpeedButton *SpeedButton_Man;
TEdit *Edit_VPrik;
TLabel *Label_VAkt;
TListBox *ListBox_Prik;
void __fastcall Ukonit1Click(TObject *Sender);
void __fastcall BitBtn_COMClick(TObject *Sender);
void __fastcall ComPortOpen(TObject *Sender);
void __fastcall ComPortClose(TObject *Sender);
void __fastcall BitBtn_RefrClick(TObject *Sender);
void __fastcall FormCreate(TObject *Sender);
void __fastcall ListBox_TstClick(TObject *Sender);
void __fastcall BitBtn_TstClick(TObject *Sender);
void __fastcall wait(double seconds);
void __fastcall BitBtn_ZarClick(TObject *Sender);
void __fastcall SpeedButton_SledClick(TObject *Sender);
void __fastcall BitBtn_AplyClick(TObject *Sender);
void __fastcall Memo_NactenDblClick(TObject *Sender);
void __fastcall BitBtn_UlozClick(TObject *Sender);
void __fastcall BitBtn_ZarSMSClick(TObject *Sender);
void __fastcall SpeedButton_NahrClick(TObject *Sender);
void __fastcall BitBtn_NahrClick(TObject *Sender);
void __fastcall BitBtnTstSMSClick(TObject *Sender);
void __fastcall Memo_SMSChange(TObject *Sender);
void __fastcall BitBtn_PITstClick(TObject *Sender);
void __fastcall PingDnsLookupDone(TObject *Sender, WORD
Error);
void __fastcall PingDisplay(TObject *Sender, TObject *Icmp,
AnsiString Msg);
void __fastcall PingEchoRequest(TObject *Sender, TObject
*Icmp);
void __fastcall PingEchoReply(TObject *Sender, TObject
*Icmp, int Status);
void __fastcall BitBtn_PIZarClick(TObject *Sender);
void __fastcall BitBtn_WTstClick(TObject *Sender);
void __fastcall SpeedButton_PomClick(TObject *Sender);
void __fastcall BitBtn_WZarClick(TObject *Sender);
void __fastcall CppWebBrowserNavigateComplete2(TObject
*Sender,
LPDISPATCH pDisp, Variant *URL);
void __fastcall CppWebBrowserDocumentComplete(TObject
*Sender,
LPDISPATCH pDisp, Variant *URL);
void __fastcall BitBtn_DTestClick(TObject *Sender);
void __fastcall HTTPGetProgress(TObject *Sender, int
TotalSize,
int Readed);
void __fastcall HTTPGetError(TObject *Sender);
void __fastcall HTTPGetDoneFile(TObject *Sender,
AnsiString FileName, int FileSize);
void __fastcall BitBtn_DZarClick(TObject *Sender);
void __fastcall BitBtn_VZarClick(TObject *Sender);
void __fastcall BitBtn_VTstClick(TObject *Sender);
void __fastcall SpeedButton_ManClick(TObject *Sender);
private:// User declarations
public: // User declarations
__fastcall TForm1(TComponent* Owner);
};

```

```

//-----
-----
extern PACKAGE TForm1 *Form1;
//-----
-----
#endif

```

## B.2 Main.cpp

```

//-----
-----

#if __BORLANDC__ == 0x520 // BCB1 is BC5.20 BCB3 is BC5.30
    #define _WINSOCKAPI_ // Prevent winsock.h from being
included
#endif

#include <vcl.h>
#pragma hdrstop

#include "Main.h"
#include "time.h"
#include "string.h"
#include "math.h"
//-----
-----

#pragma package(smart_init)
#pragma link "CPort"
#pragma link "CPortCtl"
#pragma link "RasManager"
#pragma link "Ping"
#pragma link "HttpProt"
#pragma link "SHDocVw_OCX"
#pragma link "HTTPGet"
#pragma resource "*.dfm"
TForm1 *Form1;

__di_IDispatch CurDispatch = NULL; // globalni nastaveni nacistani
WWW stranky
bool velikost_W; // globalni promena pro nastaveni velikosti okna
testu DPW
clock_t start = 0, stop; // deklarace casovych znacek
int stav1, stav2; // pomocna deklarace pro test PRD
float prenos = 1, prumpren[2]; // pomocna deklarace pro test PRD
//-----
-----

__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
-----

/* Nastaveni aplikace pri spusteni. */
void __fastcall TForm1::FormCreate(TObject *Sender)
{
    /* Zalozka Testovani. Popisky bunek - Doba sestaveni spojeni. */
    StringGrid_Ses->Cells[0][0] = "Pořadí";
    StringGrid_Ses->Cells[1][0] = "První";
}

```

```

StringGrid_Ses->Cells[2][0] = "n - 2";
StringGrid_Ses->Cells[3][0] = "n - 1";
StringGrid_Ses->Cells[4][0] = "Poslední";
StringGrid_Ses->Cells[5][0] = "Průměr";
StringGrid_Ses->Cells[0][1] = "Čas [ms]";

/* Zalozka Testovani. Popisky bunek - Doba odeslani SMS. */
StringGrid_SMS->Cells[0][0] = "Pořadí";
StringGrid_SMS->Cells[1][0] = "První";
StringGrid_SMS->Cells[2][0] = "n - 2";
StringGrid_SMS->Cells[3][0] = "n - 1";
StringGrid_SMS->Cells[4][0] = "Poslední";
StringGrid_SMS->Cells[5][0] = "Průměr";
StringGrid_SMS->Cells[0][1] = "Čas [ms]";

/* Zalozka Testovani. Popisky bunek - Doba pristupu na WWW. */
StringGrid_WWW->Cells[0][0] = "Pořadí";
StringGrid_WWW->Cells[1][0] = "První";
StringGrid_WWW->Cells[2][0] = "n - 2";
StringGrid_WWW->Cells[3][0] = "n - 1";
StringGrid_WWW->Cells[4][0] = "Poslední";
StringGrid_WWW->Cells[5][0] = "Průměr";
StringGrid_WWW->Cells[0][1] = "Čas [ms]";

/* Zalozka Testovani, test DOS. Nastaveni hodnot popisu. */
Label_PocZnn->Caption = "0";
Label_PocZpp->Caption = "0";

/* Zalozka Testovani. Nastaveni viditelnosti testu. */
GroupBox_Volani->Visible = False;
GroupBox_SMS->Visible = False;
GroupBox_PING->Visible = False;
GroupBox_WWW->Visible = False;
GroupBox_PrenosDial->Visible = False;
GroupBox_Vlastni->Visible = True;

/* Nacteni seznamu moznych Dial spojeni. */
int i = 0;
ListBox_Sez->Clear();
for (i=1; i==RasManager->Count; i++){
    ListBox_Sez->Items->Add(RasManager->Name[i-1]);
}
}
//-----
-----

/* Vlastni cekaci cyklus. */
void __fastcall TForm1::wait(double seconds)
{
    clock_t endwait;
    endwait = clock () + seconds * CLOCKS_PER_SEC;
    while (clock() < endwait) {}
}
//-----
-----

/* Nabidky programu. */
/* Slozka soubor. */
void __fastcall TForm1::Ukonit1Click(TObject *Sender)
{
    Application->Terminate();
}

```

```

}
//-----
-----

/* Zalozka Pripojeni zarizeni. */
/* Obsluha tlacitka Pripojit. */
void __fastcall TForm1::BitBtn_COMClick(TObject *Sender)
{
    if (ComPort->Connected){
        ComPort->Close();
    }
    else {
        ComPort->Open();
    }
}
//-----
-----

/* Nastaveni tlacitka Pripojit. */
void __fastcall TForm1::ComPortOpen(TObject *Sender)
{
    BitBtn_COM->Caption = "Odpojit";
}
//-----
-----

/* Nastaveni tlacitka Pripojit. */
void __fastcall TForm1::ComPortClose(TObject *Sender)
{
    BitBtn_COM->Caption = "Připojit";
}
//-----
-----

/* Obsluha tlacitka Refresh. */
void __fastcall TForm1::BitBtn_RefrClick(TObject *Sender)
{
    AnsiString Str = 0;
    AnsiString Odp = 0;
    char *OdpCh;
    char *pom;

    Memo_COPS->Lines->Clear();
    //ComPort->ClearBuffer(True, True);

    ComPort->WriteStr("ATE0\r");
    wait(0.5);
    ComPort->ReadStr(Odp, 100);
    wait(0.5);

    Str = "AT+CGMI\r";
    ComPort->WriteStr(Str);
    wait(0.5);
    ComPort->ReadStr(Odp, 100);
    wait(0.5);
    OdpCh = Odp.c_str();
    pom = strtok(OdpCh, "\r\n");
    Edit_CGMI->Text = pom;

    Str = "AT+CGMM\r";

```

```

ComPort->WriteStr(Str);
wait(0.2);
ComPort->ReadStr(Odp,100);
wait(0.2);
OdpCh = Odp.c_str();
pom = strtok(OdpCh, "\\r\\n");
Edit_CGMM->Text = pom;

Str = "AT+CGMR\\r";
ComPort->WriteStr(Str);
wait(0.2);
ComPort->ReadStr(Odp,100);
wait(0.2);
OdpCh = Odp.c_str();
pom = strtok(OdpCh, "\\r\\n");
Edit_CGMR->Text = pom;

Str = "AT+CGSN\\r";
ComPort->WriteStr(Str);
wait(0.2);
ComPort->ReadStr(Odp,100);
wait(0.2);
OdpCh = Odp.c_str();
pom = strtok(OdpCh, "\\r\\n");
Edit_CGSN->Text = pom;

Str = "AT+CIMI\\r";
ComPort->WriteStr(Str);
wait(0.2);
ComPort->ReadStr(Odp,100);
wait(0.2);
OdpCh = Odp.c_str();
pom = strtok(OdpCh, "\\r\\n");
Edit_CIMI->Text = pom;

Str = "AT+CCLK?\\r";
ComPort->WriteStr(Str);
wait(0.2);
ComPort->ReadStr(Odp,100);
wait(0.2);
OdpCh = Odp.c_str();
pom = strtok(OdpCh, "+CCLK\\r\\n");
Edit_CCLK->Text = pom;

Str = "AT+CBC\\r";
ComPort->WriteStr(Str);
wait(0.2);
ComPort->ReadStr(Odp,100);
wait(0.2);
OdpCh = Odp.c_str();
pom = strtok(OdpCh, "+CBC:\\r\\n");
Edit_CBC->Text = pom;

Str = "AT+CSQ\\r";
ComPort->WriteStr(Str);
wait(0.2);
ComPort->ReadStr(Odp,100);
wait(0.2);
OdpCh = Odp.c_str();
pom = strtok(OdpCh, "+CSQ:\\r\\n");
Edit_CSQ->Text = pom;

```

```

Str = "AT+CREG?\r";
ComPort->WriteStr(Str);
wait(0.3);
ComPort->ReadStr(Odp,100);
wait(0.2);
OdpCh = Odp.c_str();
pom = strtok(OdpCh, "\r\n");
Edit_CREG->Text = pom;

Str = "AT+COPS?\r";
ComPort->WriteStr(Str);
wait(1);
ComPort->ReadStr(Odp,100);
OdpCh = Odp.c_str();
pom = strtok(OdpCh, "\r\n");
Memo_COPS->Lines->Add(pom);

free(OdpCh);
free(pom);
}
//-----
-----

/* Zalozka Testovani. */
/* Volba zobrazeni testu. */
void __fastcall TForm1::ListBox_TstClick(TObject *Sender)
{
switch(ListBox_Tst->ItemIndex)
{
case 0 : GroupBox_Volani->Visible = True;
          GroupBox_SMS->Visible = False;
          GroupBox_PING->Visible = False;
          GroupBox_WWW->Visible = False;
          GroupBox_PrenosDial->Visible = False;
          GroupBox_Vlastni->Visible = False;
          break;
case 1 : GroupBox_Volani->Visible = False;
          GroupBox_SMS->Visible = True;
          GroupBox_PING->Visible = False;
          GroupBox_WWW->Visible = False;
          GroupBox_PrenosDial->Visible = False;
          GroupBox_Vlastni->Visible = False;
          break;
case 2 : GroupBox_Volani->Visible = False;
          GroupBox_SMS->Visible = False;
          GroupBox_PING->Visible = True;
          GroupBox_WWW->Visible = False;
          GroupBox_PrenosDial->Visible = False;
          GroupBox_Vlastni->Visible = False;
          break;
case 3 : GroupBox_Volani->Visible = False;
          GroupBox_SMS->Visible = False;
          GroupBox_PING->Visible = False;
          GroupBox_WWW->Visible = True;
          GroupBox_PrenosDial->Visible = False;
          GroupBox_Vlastni->Visible = False;
          break;
case 4 : GroupBox_Volani->Visible = False;
          GroupBox_SMS->Visible = False;
          GroupBox_PING->Visible = False;

```



```

        GroupBox_WWW->Visible = False;
        GroupBox_PrenosDial->Visible = True;
        GroupBox_Vlastni->Visible = False;
        break;
    case 5 : GroupBox_Volani->Visible = False;
            GroupBox_SMS->Visible = False;
            GroupBox_PING->Visible = False;
            GroupBox_WWW->Visible = False;
            GroupBox_PrenosDial->Visible = False;
            GroupBox_Vlastni->Visible = True;
            break;
    default : GroupBox_Volani->Visible = False;
            GroupBox_SMS->Visible = False;
            GroupBox_PING->Visible = False;
            GroupBox_WWW->Visible = False;
            GroupBox_PrenosDial->Visible = False;
            GroupBox_Vlastni->Visible = True;
            break;
    }
}
//-----

/* Obsluha tlacitka rychle volby nacteni souboru. */
void __fastcall TForm1::SpeedButton_SledClick(TObject *Sender)
{
    OpenDialog->Execute();
    Edit_Sled->Text = OpenDialog->FileName;
    Memo_Nacten->Lines->LoadFromFile(OpenDialog->FileName);
}
//-----

/* Obsluha tlacitka Aplikovat. */
void __fastcall TForm1::BitBtn_AplyClick(TObject *Sender)
{
    Memo_Test->Lines->LoadFromFile(OpenDialog->FileName);
    Memo_Nacten->Clear();
}
//-----

/* Obnoveni vypisu nacteneho souboru po dvojkliku. */
void __fastcall TForm1::Memo_NactenDbClick(TObject *Sender)
{
    Memo_Nacten->Lines->LoadFromFile(OpenDialog->FileName);
}
//-----

/* Ulozeni vytvoreneho vypisu. */
void __fastcall TForm1::BitBtn_UlozClick(TObject *Sender)
{
    SaveDialog->Execute();
    Memo_Test->Lines->SaveToFile(SaveDialog->FileName);
}
//-----

/* Test - Doba sestaveni spojeni. Obsluha tlacitka Testovat. */
void __fastcall TForm1::BitBtn_TstClick(TObject *Sender)

```

```

{
/* Testovani pripojeni k portu COM. */
if (ComPort->Connected == False){
    ShowMessage("Zařizení není připojeno k portu COM");
    PageControl->SelectNextPage(false);
}

/* Deklarace promennych. */
char *buffer;
char pombuf[42];
char *retez, *pc, *spojeni;
int handle = 0;
int delka = 0, i = 0, pocet = 0, misto = 0;
int velikost = StrToInt(Edit_Opak->Text); // pocet mereni
AnsiString Odpoved;
float *pole;
pole = (float *) malloc(velikost * sizeof(float));
float prum = 0;
clock_t start, stop; // pro mereni casu

/* Nalezeni funkce callsetup */
ComPort->WriteStr("AT+CIND=?\r"); // dotaz na CIND
wait(0.2);
ComPort->ReadStr(Odpoved,100);
retez = Odpoved.c_str(); // prevedeni Ansi retezce na char
while(strchr(retez, '\\') != NULL){ // odstraneni znaku '\\'
    i = strcspn(retez, "\\");
    retez[i] = ':';
}
while(strchr(retez, '(') != NULL){ // odstraneni znaku '('
    i = strcspn(retez, "(");
    retez[i] = ':';
}
while(strchr(retez, ')') != NULL){ // odstraneni znaku ')'
    i = strcspn(retez, ")");
    retez[i] = ':';
}
while(strchr(retez, ',') != NULL){ // odstraneni znaku ','
    i = strcspn(retez, ",");
    retez[i] = ':';
}
while(strchr(retez, '0') != NULL){ // odstraneni znaku '0'
    i = strcspn(retez, "0");
    retez[i] = ':';
}
while(strchr(retez, '-') != NULL){ // odstraneni znaku '-'
    i = strcspn(retez, "-");
    retez[i] = ':';
}
}
pc = strtok(retez, ":", 0); // hledani umistení callsetup
do {
    pc = strtok(NULL, ":", 0);
    pocet++;
    if (strcmp("callsetup", pc) == 0){
        misto = pocet;
        misto = misto/2;
        break;
    }
} while(pc != NULL);

/* Vytvoreni casoveho udaje pro txt. */

```

```

time_t rawtime;
struct tm * timeinfo;
char buffer_time [21];
time ( &rawtime );
timeinfo = localtime ( &rawtime );
strftime (buffer_time,21,"DSS%y%m%d-%H%M%S.txt",timeinfo);

/* Vytvoreni souboru a zapis hlavicky. */
handle = FileCreate(buffer_time); // vytvoreni noveho souboru
FileSeek(handle, 0, 2); // nastaveni kurzoru na prvni pozici
buffer = "TEST: Doba sestavení spojení - ";
strftime (pobuf,42,"Čas měření: %d.%m.%Y
%H:%M:%S\r\n\r\n",timeinfo);
strcat(buffer, pobuf);
delka = strlen(buffer);
FileWrite(handle, buffer, delka); // zapis do souboru
strcpy(buffer, "");
strcat(buffer, "Volané číslo: ");
strcat(buffer, (Edit_Tel->Text.c_str()));
strcat(buffer, " Počet opakování: ");
strcat(buffer, (IntToStr(velikost).c_str()));
strcat(buffer, "\r\n\r\n");
delka = strlen(buffer);
FileWrite(handle, buffer, delka); // zapis do souboru

/* Vytvoreni AT prikazu D. */
strcpy(buffer, "");
strcat(buffer, "ATD");
strcat(buffer, (Edit_Tel->Text.c_str()));
strcat(buffer, ";");

/* Merici cyklus. */
for ( i=1; i < (velikost+1); i++){
    ComPort->WriteStr(buffer); // vytoceni cisla
    start = clock(); // zacatek mereni
    wait(0.2);
    ComPort->ReadStr(Odpoved,50); // potvrzeni spojeni

    /* Testovani spravnosti spojeni. */
    if (Odpoved.AnsiPos("OK")!=0){
        spojeni = "NE";
    }
    else if (Odpoved.AnsiPos("NO CARRIER")!=0){
        spojeni = "ANO";
        goto zapis; // presun na zapis souboru
    }

    /* Testovani uskutecneni spojeni. */
    do {
        ComPort->WriteStr("AT+CIND?\r");
        wait(0.2);
        ComPort->ReadStr(Odpoved,50);
        retez = Odpoved.c_str();
        pc = 0;
        pc = strtok(retez, ",", "");
        for (i=1; i != misto; i++){
            pc = strtok(NULL, ",", "");
        }
    } while((pc, "3"));
    ComPort->WriteStr("ATH\r"); // ukonceni spojeni
    zapis: // navesti pri chybe

```

```

    stop = clock(); // konec mereni
    wait(0.2);
    ComPort->ReadStr(Odpoved,50);
    pole[i-1] = ((stop - start)/((double) CLK_TCK))*100; // zmerena
hodnota

    /* Vypocet prumeru. */
    if(i==1){
        prum = pole[0];
    }
    else {
        prum = (prum + pole[i-1])/2;
    }

    /* Vypis prehledovych hodnot. */
    StringGrid_Ses->Cells[1][1] = FloatToStr(pole[0]);
    StringGrid_Ses->Cells[2][1] = FloatToStr(pole[i-2]);
    StringGrid_Ses->Cells[3][1] = FloatToStr(pole[i-1]);
    StringGrid_Ses->Cells[4][1] = FloatToStr(pole[i]);
    StringGrid_Ses->Cells[5][1] = FloatToStr(prum);

    /* Zapis vyslednych hodnot do souboru. */
    strcpy(buffer, (IntToStr(i).c_str()));
    strcat(buffer, ". měření: ");
    strcat(buffer, FloatToStr(pole[i-1]).c_str());
    strcat(buffer, "ms - Průměr: ");
    strcat(buffer, FloatToStr(prum).c_str());
    strcat(buffer, " Chyba spojení: ");
    strcat(buffer, spojeni);
    strcat(buffer, "\r\n");
    delka = strlen(buffer);
    FileWrite(handle, buffer, delka);
}

/* Ukonceni souboru a uvolneni pameti. */
FileClose(handle);
free((void *) pole);
free(buffer);
free(retez);
free(pc);
free(spojeni);
}
//-----
-----

/* Obsluha tlacitka Zaradit. Test DSS. */
void __fastcall TForm1::BitBtn_ZarClick(TObject *Sender)
{
    Memo_Test->Lines->Add("DSS - Doba sestavení spojení. Opakování: "
+ Edit_Opak->Text + "x. Tel. číslo: " + Edit_Tel->Text);
}
//-----
-----

/* Test - Doba odeslání SMS. Obsluha tlacitka Testovat. */
void __fastcall TForm1::BitBtn_TstSMSClick(TObject *Sender)
{
    /* Testovani pripojeni k portu COM. */
    if (ComPort->Connected == False){
        ShowMessage("Zařízení není připojeno k portu COM");
        PageControl->SelectNextPage(false);
    }
}

```

```

}

/* Deklarace promennych. */
char *buffer;
char pobuf[42];
int handle = 0, delka = 0;
int velikost = StrToInt(Edit_PocOpak->Text); // pocet mereni
AnsiString Odpoved;
float *pole;
pole = (float *) malloc(velikost * sizeof(float));
float prum = 0;
clock_t start, stop; // pro mereni casu

/* Vytvoreni casoveho udaje pro txt. */
time_t rawtime;
struct tm * timeinfo;
char buffer_time [21];
time ( &rawtime );
timeinfo = localtime ( &rawtime );
strftime (buffer_time,21,"DOS%y%m%d-%H%M%S.txt",timeinfo);

/* Vytvoreni souboru a zapis hlavicky. */
handle = FileCreate(buffer_time); // vytvoreni souboru
FileSeek(handle, 0, 2); // nastaveni kurzoru na prvni pozici
buffer = "TEST: Doba odesláni SMS - ";
strftime (pobuf,42,"Čas měření: %d.%m.%Y
%H:%M:%S\r\n\r\n",timeinfo);
strcat(buffer, pobuf);
delka = strlen(buffer);
FileWrite(handle, buffer, delka); // zapis do souboru
strcpy(buffer, "");
strcat(buffer, "Volané číslo: ");
strcat(buffer, (Edit_TelCis->Text.c_str()));
strcat(buffer, " Počet opakování: ");
strcat(buffer, (IntToStr(velikost).c_str()));
strcat(buffer, "\r\n\r\n");
strcat(buffer, (Memo_SMS->Text.c_str()));
strcat(buffer, "\r\n\r\n");
delka = strlen(buffer);
FileWrite(handle, buffer, delka); // zapis do souboru

/* Merici cyklus. */
for (int i=1; i < (velikost+1); i++)
{
    ComPort->WriteStr(buffer); // zapis prikazu
    start = clock(); // zacatek mereni
    wait(0.2);
    ComPort->ReadStr(Odpoved,50); // potvrzeni odpovedi

    stop = clock(); // konec mereni
    pole[i-1] = ((stop - start)/((double) CLK_TCK))*100;

    /* Vypocet prumeru. */
    if(i==1){
        prum = pole[0];
    }
    else {
        prum = (prum + pole[i-1])/2;
    }

    /* Vypis prehledovych hodnot. */

```

```

StringGrid_SMS->Cells[1][1] = FloatToStr(pole[0]);
StringGrid_SMS->Cells[2][1] = FloatToStr(pole[i-2]);
StringGrid_SMS->Cells[3][1] = FloatToStr(pole[i-1]);
StringGrid_SMS->Cells[4][1] = FloatToStr(pole[i]);
StringGrid_SMS->Cells[5][1] = FloatToStr(prum);

/* Zapis vyslednych hodnot do souboru. */
strcpy(buffer, (IntToStr(i).c_str()));
strcat(buffer, ". měření: ");
strcat(buffer, FloatToStr(pole[i-1]).c_str());
strcat(buffer, "ms - Průměr: ");
strcat(buffer, FloatToStr(prum).c_str());
strcat(buffer, " Chyba spojení: \r\n");
delka = strlen(buffer);
FileWrite(handle, buffer, delka);
}

/* Ukonceni souboru a uvolneni pameti. */
FileClose(handle);
free((void *) pole);
free(buffer);
}
//-----
-----

/* Obsluha tlacitka Zaradit. Test DOS. */
void __fastcall TForm1::BitBtn_ZarSMSClick(TObject *Sender)
{
Memo_Test->Lines->Add("DOS - Doba odeslání SMS. Opakování: " +
Edit_PocOpak->Text + "x. Tel. číslo: " + Edit_TelCis->Text + " SMS
zpráva: " + Memo_SMS->Text);
}
//-----
-----

/* Obsluha tlacitka rychle volby nacteni souboru. Test DOS. */
void __fastcall TForm1::SpeedButton_NahrClick(TObject *Sender)
{
OpenDialog->Execute();
Edit_Nahr->Text = OpenDialog->FileName;
}
//-----
-----

/* Obsluha tlacitka Nacist. Test DOS. */
void __fastcall TForm1::BitBtn_NahrClick(TObject *Sender)
{
Memo_SMS->Clear();
Memo_SMS->Lines->LoadFromFile(OpenDialog->FileName);
}
//-----
-----

/* Obsluha Memo_SMS pri zmene. */
void __fastcall TForm1::Memo_SMSChange(TObject *Sender)
{
int pocet = 0;
Label_PocZnn->Caption = Memo_SMS->GetTextLen();
pocet = Memo_SMS->GetTextLen();
if (pocet == 800){
pocet = 5;
}
}

```

```

    }
    else if ((pocet/160) > 1){
        pocet = ((pocet/160)+1);
    }
    else if (pocet == 0){
        pocet = 0;
    }
    else if ((pocet/160) == 0){
        pocet = 1;
    }
    Label_PocZpp->Caption = IntToStr(pocet);
}
//-----
-----

/* Test - PING test. Obsluha tlacitka Testovat. */
void __fastcall TForm1::BitBtn_PITstClick(TObject *Sender)
{
    if (BitBtn_PITst->Caption == "Stop"){
        Ping->CancelDnsLookup();
    }
    BitBtn_PITst->Caption = "Stop";

    /* Testovani pripojeni k Dial-up. */
    /* if (ComPort->Connected == False){
        ShowMessage("Zařizeni není připojeno k Dial-up");
        PageControl->SelectNextPage(false);
    } */

    /* Nastavení hodnot procesu. */
    Ping->Size = StrToInt(Edit_PIVel->Text); // velikost
    Ping->TTL = StrToInt(Edit_PITime->Text); // Time To Leave
    Ping->Timeout = StrToInt(Edit_PIOut->Text); // Timeout

    Memo_PI->Clear(); // vycistení záznamu
    Ping->DnsLookup(Edit_PIHost->Text); // volání procesu pro PING

    BitBtn_PITst->Caption = "Testovat";
}
//-----
-----

/* Funkce PING */
void __fastcall TForm1::PingDnsLookupDone(TObject *Sender, WORD
Error)
{
    /* Deklarace promenných. */
    char *buffer;
    char pombuf[42];
    int handle = 0, delka = 0;
    unsigned int min = 1000, max = 0, err = 0;
    int prum = 0, ztr = 0;

    /* Zapis první části. */
    Memo_PI->Lines->Add("Přikaz PING na " + Edit_PIHost->Text + " ["
+ Ping->DnsResult + "] s délkou " + IntToStr(Ping->Size) + "
bajtů:");
    Memo_PI->Lines->Add("");

    /* Kontrola existence adresy. */
    if (Error != 0) {

```

```

    Memo_PI->Lines->Add("Neznámý cíl " + Edit_PiHost->Text);
    return;
}

/* Testovací smyčka. */
for (int i=1; i <= StrToInt(Edit_PiOpak->Text); i++){
    Ping->Address = Ping->DnsResult; //DNS
    Ping->Ping(); // testování
    if (Ping->Reply.Status != 0) { // hledání erroru
        err ++;
    }
    if (Ping->Reply.RTT > max){ // vyhodnocení času měření
        max = Ping->Reply.RTT;
    }
    else if (Ping->Reply.RTT < min){
        min = Ping->Reply.RTT;
    }
    prum += Ping->Reply.RTT;
}

prum = prum/StrToInt(Edit_PiOpak->Text); // výpočet průměru
ztr = (err/StrToInt(Edit_PiOpak->Text))*100; // výpočet
ztracených

/* Zapsí vyhodnocení. */
Memo_PI->Lines->Add("");
Memo_PI->Lines->Add("Statistika ping pro " + Ping->DnsResult +
":");
Memo_PI->Lines->Add("Pakety: Odeslané = " + Edit_PiOpak->Text +
", Přijaté = " + IntToStr(StrToInt(Edit_PiOpak->Text) - err) + ",
Ztracené = " + IntToStr(err) + " (ztráta " + IntToStr(ztr) +
"%),");
Memo_PI->Lines->Add("Přibližná doba do přijetí
odezvy v milisekundách:");
Memo_PI->Lines->Add("\tMinimum = " + IntToStr(min) + "ms, Maximum
= " + IntToStr(max) + "ms, Průměr = " + IntToStr(prum) + "ms");

/* Vytvoření časového údaje pro txt. */
time_t rawtime;
struct tm * timeinfo;
char buffer_time [21];
time ( &rawtime );
timeinfo = localtime ( &rawtime );
strftime (buffer_time,21,"PIN%y%m%d-%H%M%S.txt",timeinfo);

/* Vytvoření souboru a zápis hlavičky. */
handle = FileCreate(buffer_time); // vytvoření souboru
FileSeek(handle, 0, 2); // nastavení kurzoru na první pozici
buffer = "TEST: PING - ";
strftime (pobuf,42,"Čas měření: %d.%m.%Y
%H:%M:%S\r\n\r\n",timeinfo);
strcat(buffer, pobuf);
delka = strlen(buffer);
FileWrite(handle, buffer, delka); // zápis do souboru
strcpy(buffer, "");
strcat(buffer, "Host: ");
strcat(buffer, (Edit_PiHost->Text.c_str()));
strcat(buffer, " Počet opakování: ");
strcat(buffer, (Edit_PiOpak->Text.c_str()));
strcat(buffer, "\r\n\r\n");
strcat(buffer, (Memo_PI->Text.c_str()));

```



```

    delka = strlen(buffer);
    FileWrite(handle, buffer, delka); // zapis do souboru
    FileClose(handle); // uzavreni souboru
    free(buffer);
}
//-----
-----

    /* Vypis zprav z PING. */
void __fastcall TForm1::PingDisplay(TObject *Sender, TObject *Icmp,
AnsiString Msg)
{
    Memo_PI->Lines->Add(Msg);
}
//-----
-----

    /* Odesilani dotazu a zapis hodnot. */
void __fastcall TForm1::PingEchoRequest(TObject *Sender, TObject
*Icmp)
{
    Memo_PI->Lines->Add("Odesílám na " + Ping->HostName + " (" +
Ping->HostIP + "): bajty=" + IntToStr(Ping->Size) + " TTL=" +
IntToStr(Ping->TTL));
}
//-----
-----

    /* Vyhodnoceni odpovedi a zapis vysledku. */
void __fastcall TForm1::PingEchoReply(TObject *Sender, TObject
*Icmp, int Error)
{
    if (Error == 0){
        Memo_PI->Lines->Add("Nelze provést ping na adresu (" + Ping-
>HostIP + ") : " + Ping->ErrorString);
    }
    else {
        Memo_PI->Lines->Add("Odpověď od " + Ping->HostName + " (" +
Ping->HostIP + "): bajty=" + IntToStr(Ping->Size) + " čas=" +
IntToStr(Ping->Reply.RTT) + "ms TTL=" + IntToStr(Ping->TTL));
    }
}
//-----
-----

    /* Obsluha tlacitka Zaradit. Test PIN. */
void __fastcall TForm1::BitBtn_PIZarClick(TObject *Sender)
{
    Memo_Test->Lines->Add("PING - PING. Opakování: " + Edit_PIOpak-
>Text + "x. Adresa: " + Edit_PIHost->Text + " TTL: " + Edit_PITime-
>Text + " Velikost: " + Edit_PIVel->Text + " Timeout: " +
Edit_PIOut->Text);
}
//-----
-----

    /* Test - Doba pristupu na WWW. Obsluha tlacitka Testovat. */
void __fastcall TForm1::BitBtn_WTstClick(TObject *Sender)
{
    /* Deklarace promennych. */
    char *adr, *pch;

```

```

//char *spojeni; // urceni zda stranka byla nalezena
char *buffer;
char pombuf[42];
int handle = 0, delka = 0;
float prum = 0;
float *pole;
int velikost = Memo_WSez->Lines->Count;
pole = (float *) malloc(velikost * sizeof(float));
clock_t start, stop; // pro mereni casu

/* Vytvoreni casoveho udaje pro txt. */
time_t rawtime;
struct tm * timeinfo;
char buffer_time [21];
time ( &rawtime );
timeinfo = localtime ( &rawtime );
strftime (buffer_time,21,"DPW%y%m%d-%H%M%S.txt",timeinfo);

/* Vytvoreni souboru a zapis hlavicky. */
handle = FileCreate(buffer_time); // vytvoreni souboru
FileSeek(handle, 0, 2); // nastaveni kurzoru na prvni pozici
buffer = "TEST: Doba přístupu na WWW - ";
strftime (pombuf,42,"Čas měření: %d.%m.%Y
%H:%M:%S\r\n\r\n",timeinfo);
strcat(buffer, pombuf);
delka = strlen(buffer);
FileWrite(handle, buffer, delka); // zapis do souboru
strcpy(buffer, "");
strcat(buffer, (Memo_WSez->Text.c_str()));
strcat(buffer, "\r\n");
delka = strlen(buffer);
FileWrite(handle, buffer, delka); // zapis do souboru

/* Testovací smyčka. */
adr = Memo_WSez->Text.c_str(); // retezec s adresami
pch = strtok(adr, "\r\n"); // rozklad retezce na jednotlivé adresy
for (int i=1; i <= (Memo_WSez->Lines->Count); i++){
    Edit_Pom->Text = pch; // pomoc k převedení char na WideString
    pch = strtok(NULL, "\r\n");
    start = clock(); // začátek měření
    CppWebBrowser->Navigate(WideString(Edit_Pom->Text));
    stop = clock(); // konec měření
    Sleep(1000);
    pole[i-1] = ((stop - start)/((double) CLK_TCK));

    /* Vypocet prumeru. */
    if(i==1){
        prum = pole[0];
    }
    else {
        prum = (prum + pole[i-1])/2;
    }

    /* Vypis prehledovych hodnot. */
    StringGrid_WWW->Cells[1][1] = FloatToStr(pole[0]);
    StringGrid_WWW->Cells[2][1] = FloatToStr(pole[i-2]);
    StringGrid_WWW->Cells[3][1] = FloatToStr(pole[i-1]);
    StringGrid_WWW->Cells[4][1] = FloatToStr(pole[i]);
    StringGrid_WWW->Cells[5][1] = FloatToStr(prum);

    /* Zapis vyslednych hodnot do souboru. */

```

```

        strcpy(buffer, (IntToStr(i).c_str()));
        strcat(buffer, ". měření: ");
        strcat(buffer, FloatToStr(pole[i-1]).c_str());
        strcat(buffer, "ms - Průměr: ");
        strcat(buffer, FloatToStr(prum).c_str());
        //strcat(buffer, " Chyba spojení: ");
        //strcat(buffer, spojeni);
        strcat(buffer, "\r\n");
        delka = strlen(buffer);
        FileWrite(handle, buffer, delka);
    }

    /* Ukonceni souboru a uvolneni pameti. */
    FileClose(handle);
    free((void *) pole);
    free(buffer);
    free(adr);
    free(pch);
    //free(spojeni);
}
//-----
-----

    /* Priznak vyhledani zadane stranky. Test DPW. */
void __fastcall TForm1::CppWebBrowserNavigateComplete2(TObject
*Sender,
    LPDISPATCH pDisp, Variant *URL)
{
    if (!CurDispatch){
        CurDispatch = pDisp; // ulozeni pro porovnani
    }
}
//-----
-----

    /* Priznak nacteni zadane stranky. Test DPW. */
void __fastcall TForm1::CppWebBrowserDocumentComplete(TObject
*Sender,
    LPDISPATCH pDisp, Variant *URL)
{
    if (pDisp == CurDispatch){
        Beep(); // nahrani dokumentu, jeste ne ramcu
        CurDispatch = NULL; // vynulovani globalni promenne
    }
}
//-----
-----

    /* Obsluha pomocneho tlacitka. Test DPW. */
void __fastcall TForm1::SpeedButton_PomClick(TObject *Sender)
{
    /* Zmena velikosti a usporadani ovladani. */
    if (velikost_W == TRUE){
        CppWebBrowser->Top = 226;
        CppWebBrowser->Left = 10;
        CppWebBrowser->Height = 149;
        CppWebBrowser->Width = 382;
        BitBtn_WTst->Top = 385;
        BitBtn_WTst->Left = 224;
        BitBtn_WZar->Top = 385;
        BitBtn_WZar->Left = 312;
    }
}

```

```

        SpeedButton_Pom->Top = 384;
        SpeedButton_Pom->Left = 10;
        GroupBox_WWW->Top = 10;
        GroupBox_WWW->Left = 220;
        GroupBox_WWW->Height = 423;
        GroupBox_WWW->Width = 402;
        velikost_W = FALSE;
    }
    else if (velikost_W == FALSE){
        GroupBox_WWW->Top = 10;
        GroupBox_WWW->Left = 220;
        GroupBox_WWW->Height = 558;
        GroupBox_WWW->Width = 762;
        BitBtn_WTst->Top = 38;
        BitBtn_WTst->Left = 402;
        BitBtn_WZar->Top = 73;
        BitBtn_WZar->Left = 402;
        SpeedButton_Pom->Top = 38;
        SpeedButton_Pom->Left = 487;
        CppWebBrowser->Top = 226;
        CppWebBrowser->Left = 10;
        CppWebBrowser->Height = 322;
        CppWebBrowser->Width = 742;
        velikost_W = TRUE;
    }
}
//-----
-----

/* Obsluha tlacitka Zaradit. Test DPW. */
void __fastcall TForm1::BitBtn_WZarClick(TObject *Sender)
{
    int i;
    char *adr = Memo_WSez->Text.c_str();

    while(strchr(adr, '\n') != NULL){ // odstraneni znaku '\n'
        i = strcspn(adr, "\n");
        adr[i] = ',';
    }
    Memo_Test->Lines->Add("DPW - Doba přístupu na WWW. Testované
stránky: " + AnsiString(adr));
    free(adr);
}
//-----
-----

/* Test - Prenosova rychlost (Dial-up). Obsluha tlacitka
Testovat. */
void __fastcall TForm1::BitBtn_DTestClick(TObject *Sender)
{
    /* Nastaveni informacnich zobrazovaciu. */
    Label_DFul->Visible = False; // informace - Stahovani kompletni.
    Label_DErr->Visible = False; // informace - Chyba pri stahovani.
    Label_DStat->Visible = True; // informace o prubehu stahovani

    /* Zacatek stahovaciho procesu - Dialog pro ulozeni. */
    if (SaveDialog_Dial->Execute()){ // vyvolani dialogu pro ulozeni
        HTTPGet->URL = Edit_DURL->Text; // nastaveni adresy souboru ke
stazeni
        HTTPGet->FileName = SaveDialog_Dial->FileName;
        HTTPGet->GetFile(); // zacatek procesu stahovani souboru

```

```

    }
}
//-----
-----

/* Preces stahovani souboru. */
void __fastcall TForm1::HTTPGetProgress(TObject *Sender, int
TotalSize,
int Readed)
{
/* Deklarace promennych. */
float cas;

/* Vypocet prenosove rychlosti a doby do konce stahovani. */
if (start == 0){
stav1 = Readed; // velikost souboru pri zacatku mereneho
intervalu
start = clock(); // cas pri zacatku mereneho intervalu
}
if (((stop-start)/(double)CLK_TCK) >= 1){
prenos = (stav2-stav1); // prenosova rychlost
prumpren[0] += prenos; // ulozeni prenosove rychlosti pro
vypocet prumeru
prumpren[1] += 1; // ulozeni cyklu pro vypocet prumeru
stav1 = Readed;
start = clock();
}
else {
stav2 = Readed; // velikost souboru na konci mereneho intervalu
stop = clock(); // cas na konci mereneho intervalu
}
cas = (TotalSize-Readed)/prenos; // vypocet zbyvajiciho casu
stahovani

/* Nastaveni zobrazovacich informaci. */
ProgressBar->Max = TotalSize; // max velikost ukazatele
ProgressBar->Position = Readed; // pozice v ukazateli
Label_DVell->Caption = IntToStr(TotalSize) + " bajtů"; //
velikost souboru
Label_DPrenn->Caption = FloatToStr(prenos) + " B/s"; // prenosova
rychlost
Label_DTimm->Caption = FloatToStr(cas) + " s"; // zbyvajici cas
stahovani
Label_DStat->Caption = "Staženo " + IntToStr(Readed) + "
bajtů z " + IntToStr(TotalSize) + " " +
FloatToStr(floor(100*Readed/TotalSize)) + "%";
// informace o prubehu stahovani
}
//-----
-----

/* Chybove zobrazeni. */
void __fastcall TForm1::HTTPGetError(TObject *Sender)
{
Label_DErr->Visible = True;
}
//-----
-----

/* Obsluha po dokonceni stahovani souboru. */
void __fastcall TForm1::HTTPGetDoneFile(TObject *Sender,

```

```

        AnsiString FileName, int FileSize)
{
    /* Nastaveni informacnich zobrazovaciu. */
    Label_DStat->Visible = False;
    Label_DFul->Visible = True;
    start = 0;

    /* Deklarace promennych. */
    char *buffer;
    char pobuf[42];
    int handle = 0, delka = 0;
    float prum;
    prum = prumpren[0]/prumpren[1];

    /* Vytvoreni casoveho udaje pro txt. */
    time_t rawtime;
    struct tm * timeinfo;
    char buffer_time [21];
    time ( &rawtime );
    timeinfo = localtime ( &rawtime );
    strftime (buffer_time,21,"PRD%y%m%d-%H%M%S.txt",timeinfo);

    /* Vytvoreni souboru a zapis hlavicky. */
    handle = FileCreate(buffer_time); // vytvoreni souboru
    FileSeek(handle, 0, 2); // nastaveni kurzoru na prvni pozici
    buffer = "TEST: Přenosová rychlost (Dial-up) - ";
    strftime (pobuf,42,"Čas měření: %d.%m.%Y
%H:%M:%S\r\n\r\n",timeinfo);
    strcat(buffer, pobuf);
    delka = strlen(buffer);
    FileWrite(handle, buffer, delka); // zapis do souboru
    strcpy(buffer, "");
    strcat(buffer, "URL: ");
    strcat(buffer, Edit_DURL->Text.c_str());
    strcat(buffer, " \r\n\r\nPrůměrná přenosová rychlost: ");
    strcat(buffer, FloatToStr(prum).c_str());
    strcat(buffer, " B/s");
    delka = strlen(buffer);
    FileWrite(handle, buffer, delka); // zapis do souboru

    /* Ukonceni souboru a uvolneni pameti. */
    FileClose(handle);
    free(buffer);
}
//-----
-----

    /* Obsluha tlacitka Zaradit. Test PRD. */
void __fastcall TForm1::BitBtn_DZarClick(TObject *Sender)
{
    Memo_Test->Lines->Add("PRD - Přenosová rychlost (Dial-up). URL: "
+ Edit_DURL->Text);
}
//-----
-----

    /* Test - Vlastni test.... Obsluha tlacitka Testovat. */
void __fastcall TForm1::BitBtn_VTstClick(TObject *Sender)
{
    AnsiString Odp;

```

```

if (CheckBox_Vl->Checked == False){
    for (int i = 0; i <= ListBox_Prik->Items->Count; i++) {
        Edit_VPrik->Text = ListBox_Prik->Items->Strings[0];
        ListBox_Prik->Items->Delete(0);
        ComPort->WriteStr(Edit_VPrik->Text);
        wait(0.2);
        ComPort->ReadStr(Odp,100);
        Memo_Vypis->Text = Odp;
    }
}
else {
    Memo_Vypis->Lines->Add("Je zapnut manuální režim.");
}
}
//-----
-----

/* Obsluha krokovacího tlačítka. Test VLT. */
void __fastcall TForm1::SpeedButton_ManClick(TObject *Sender)
{
    AnsiString Odp;

    if (CheckBox_Vl->Checked == True){
        Edit_VPrik->Text = ListBox_Prik->Items->Strings[0];
        ListBox_Prik->Items->Delete(0);
        ComPort->WriteStr(Edit_VPrik->Text);
        wait(0.2);
        ComPort->ReadStr(Odp,100);
        Memo_Vypis->Text = Odp;
    }
    else {
        Memo_Vypis->Lines->Add("Není zapnut manuální režim.");
    }
}
//-----
-----

/* Obsluha tlačítka Zaradit. Test VLT. */
void __fastcall TForm1::BitBtn_VZarClick(TObject *Sender)
{
    //ListBox_Prik->Lines[0] =
    //Memo_Test->Lines->Add("VLT - Vlastní test.... Příkazy: " +
    ListBox_Prik->Lines);
}
//-----
-----

void __fastcall TForm1::ListBox_VybClick(TObject *Sender)
{
    Table->TableName = ListBox_Vyb->Items->Strings[ListBox_Vyb->ItemIndex];
    if(Table->TableName == "ATcommands.DB") {

    }
    else {

    }
}
//-----
-----

```

```
void __fastcall TForm1::BitBtn_ZadClick(TObject *Sender)
{
    ListBox_Prik->Items->Add(DBEdit_AT->Text);
}
//-----
-----
```